Baseline: Practical Control Variates for Agent Evaluation in Zero-Sum Domains

Joshua Davidson, Christopher Archibald and Michael Bowling
{joshuad, archibal, bowling}@ualberta.ca
Department of Computing Science
University of Alberta
Edmonton, AB, Canada T6G 2E8

ABSTRACT

Agent evaluation in stochastic domains can be difficult. The commonplace approach of Monte Carlo evaluation can involve a prohibitive number of simulations when the variance of the outcome is high. In such domains, variance reduction techniques are necessary, but these techniques require careful encoding of domain knowledge. This paper introduces *baseline* as a simple approach to creating low variance estimators for zero-sum multi-agent domains with high outcome variance. The baseline method leverages the self play of any available agent to produce a control variate for variance reduction, subverting any extra complexity inherent with traditional approaches. The baseline method is also applicable in situations where existing techniques either require extensive implementation overhead or simply cannot be applied. Experimental variance reduction results are shown for both cases using the baseline method. Baseline is shown to surpass state-of-the-art techniques in three-player computer poker and is competitive in two-player computer poker games. Baseline also shows variance reduction in human poker and in a mock Ad Auction tournament from the Trading Agent Competition, domains where variance reduction methods are not typically employed.

Categories and Subject Descriptors

G.3 [**Mathematics of Computing**]: Probability and Statistics—*Experimental Design*; I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems—*Games*

General Terms

Measurement, Performance, Experimentation

Keywords

Control Variates; Agent Evaluation

1. INTRODUCTION

In games, as well as many other domains where there are both aspects of chance and skill, one common problem that arises in evaluation is the ability to estimate how much of the utility each player earns is due to luck, and how much is due to skill. A simple approach is to collect a large number of independent samples of the outcome and measure the mean and standard error of the mean. This however is not ideal for many reasons, the most obvious perhaps being that it may take an impractical number of simulations to separate evenly matched opponents with any sort of statistical significance. More sophisticated approaches tackle this problem by first trying to reduce the variance in the outcome before performing any such evaluation, and are quite necessary in domains with high outcome variance. Existing variance reduction techniques often require hand crafting and substantial domain knowledge in order to be of any use. In this paper, we present a practical method for reducing the variance in multi-agent evaluation.

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013), Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.*

2. MOTIVATION

When evaluating agents in domains with high outcome variance, variance reduction techniques are crucial in situations where simply gathering more data is highly infeasible or may even impossible. The game of poker provides a prime example of this type of situation. In poker, gathering the necessary sample size to simply use the raw outcome would be a near impossible feat for human players. For instance, if two players are of similar skill, it may require tens, or even hundreds, of thousands of hands of play in order to naively evaluate who is the better player. Even insanely fast human players, making thousands of decisions an hour playing for ten hours a day would take hundreds of days to generate this amount of data. Additionally, over such long timeframes, the assumption that the samples are independent and identically distributed probably becomes questionable.

Analogous to this is the case where one would like to evaluate a computer agent's skill against that of a professional human player. Much as in the case of human versus human evaluation, the need to reduce the number of samples required for human versus computer evaluation is crucial since the cost of gathering data remains extremely high. We are either required to disregard significance (which is not uncommon), or be more clever.

One variance reduction technique employed in computer poker is a method known as *duplicate*. Duplicate requires each player to play every hand multiple times, once for each possible configuration of players at the table. This method is practical and effective for computer programs since they have the ability to easily forget what happened in the past

and is in fact used in the Annual Computer Poker Competition (ACPC) [1]. One drawback of the duplicate method is the question of effectiveness as the number of agents in a domain increases since the number of samples needed for one data point is factorial in the number of agents. The duplicate method is also problematic in matches involving humans. Human players cannot be expected to forget pervious hands they may have played, especially in the small time frames required to effectively hold a competition.

Nevertheless, duplicate has been used to evaluate teams of human poker players by averaging the winnings of each hand for all the players on a team. Duplicate was used in the previous man versus machine poker competitions [10] as well as a six player limit Texas hold 'em human competition hosted by the International Federation of Poker (IFP) [4]. As a simplification, the IFP human duplicate competition only used six players per team instead of the 720 required to do a full duplicate match. The IFP competition further increased the complexity of running the competition by using physical cards and tables. This choice necessitated the pre-shuffling of 432 decks of cards for just 72 hands of play at each of the six final tables. The fragility of a duplicate match on this scale was reinforced when one of the cards was overturned prematurely during a hand on one of the tables. This resulted in that hand being stricken from the results on all six tables as it can no longer be used in the duplicate analysis.

It is these types of situations that motivated the development of the *baseline* approach to variance reduction, a simple method for creating practical low variance estimators without the need for special domain knowledge.

3. BACKGROUND

This section gives a brief overview of existing traditional approaches to variance reduction. Also provided in this section are some examples of systems that implement variations of these techniques.

3.1 General Methods for Variance Reduction

Several general methods have been developed in the statistics literature for reducing the variance inherent in estimating random variables [7]. In what follows, assume that the random variable of interest is X and that there are n independent and identically distributed samples of X, $(X_1, X_2, \ldots, X_n)$. The goal is to estimate $\mathbb{E}[X]$, provided it exists, from these n samples, and to do so with as little variance as possible. We denote by $\bar{X} = \frac{1}{n}(X_1 + \cdots + X_n)$ the *sample mean*. Assuming that $\mathbb{E}[X]$ exists, $\bar{X}$ is an unbiased estimator of $\mathbb{E}[X]$. The goal of each of the techniques presented is to give another estimator of $\mathbb{E}[X]$ with hopefully lower variance than $\bar{X}$

3.1.1 Control Variates

Control variates are one effective way to reduce the error of Monte Carlo simulations [3]. Assume that we are given some additional statistic Y that is correlated with X, and also that $\mu_Y = \mathbb{E}[Y]$ is a known and existing quantity. We can then construct an improved estimate of $\mathbb{E}[X]$ by considering a new random variable $Z := X + c(Y - \mathbb{E}[Y])$. Computing the sample mean of Z, $\bar{Z}$, is then equal to

$$\begin{aligned}\bar{Z} &= \bar{X} + c(\bar{Y} - \mathbb{E}[Y]) \\ &= \frac{1}{n}\sum_{i=1}^{n}(X_i + c(Y_i - \mathbb{E}[Y]))\end{aligned} \tag{1}$$

It is then the case that $\bar{Z}$ is an unbiased estimator of $\mathbb{E}[X]$ for any $c \in \mathbb{R}$ since

$$\begin{aligned}\mathbb{E}[Z] &= \mathbb{E}[X + c(Y - \mathbb{E}[Y])] \\ &= \mathbb{E}[X] + \mathbb{E}[c(Y - \mathbb{E}[Y])] \\ &= \mathbb{E}[X] + c(\mathbb{E}[Y] - \mathbb{E}[Y]) \\ &= \mathbb{E}[X]\end{aligned} \tag{2}$$

In such a setting, Y is referred to as the *control variate*. The variance of Z can then be computed as

$$\begin{aligned}\mathbb{V}\mathrm{ar}[Z] &= \mathbb{V}\mathrm{ar}[X + c(Y - \mathbb{E}[Y])] \\ &= \mathbb{V}\mathrm{ar}[X] + c^2\mathbb{V}\mathrm{ar}[Y] + 2c\mathbb{C}\mathrm{ov}[X, Y]\end{aligned} \tag{3}$$

and this variance is minimized by choosing the optimal coefficient c^* by

$$c^* = -\frac{\mathbb{C}\mathrm{ov}[X, Y]}{\mathbb{V}\mathrm{ar}[X]} \tag{4}$$

Resulting in a final variance of

$$\begin{aligned}\mathbb{V}\mathrm{ar}[Z] &= \mathbb{V}\mathrm{ar}[X] - \frac{\mathbb{C}\mathrm{ov}[X, Y]^2}{\mathbb{V}\mathrm{ar}[Y]} \\ &= (1 - \rho_{X,Y}^2)\mathbb{V}\mathrm{ar}[X]\end{aligned} \tag{5}$$

where $\rho_{X,Y}$ is the correlation between X and Y. This shows that the amount of reduction in the variance of an estimate when using this technique is highly dependent on the control variate Y being strongly correlated with the original random variable X.

3.1.2 Multiple Control Variates

The natural extension of the aforementioned formulas to the case of *multiple control variates* is to have Y instead be a vector of statistics correlated with X, denoted as $\vec{Y}$. We can define a new random variable M as

$$M = X - \vec{c}^{\top}(\vec{Y} - \mathbb{E}[\vec{Y}]) \tag{6}$$

and the optimal coefficient vector $\vec{c^*}$ can be computed, similarly to before, as

$$\vec{c}^* = \Sigma_{\vec{Y}\vec{Y}}^{-1}\Sigma_{X\vec{Y}} \tag{7}$$

were $\Sigma_{X\vec{Y}}$ is the vector of covariances between X and $\vec{Y}$ and $\Sigma_{\vec{Y}\vec{Y}}$ is the covariance matrix of $\vec{Y}$.

3.1.3 Antithetic Variates

Antithetic variates are yet another technique for reducing the variance when estimating $\mathbb{E}[X]$. Assume we have two identically distributed random variables, X and Y, and consider their average W as an estimate for $\mathbb{E}[X]$ as

$$W = \frac{X + Y}{2}$$

The variance of W is equal to

$$\begin{aligned}\mathbb{V}\mathrm{ar}(W) &= \frac{1}{4}(\mathbb{V}\mathrm{ar}[X] + \mathbb{V}\mathrm{ar}[Y]) + \frac{1}{2}\mathbb{C}\mathrm{ov}[X, Y] \\ &= \frac{\mathbb{V}\mathrm{ar}[X]}{2} + \frac{1}{2}\mathbb{C}\mathrm{ov}[X, Y]\end{aligned} \tag{8}$$

This expression can be exploited to reduce the variance of W by having X and Y be negatively correlated with each other. The antithetic variates method works on the premise that for any given sample $X_i \sim X$, an antithesis Y_i can be generated in such a way that there is strong negative correlation between the two values. Y must be carefully selected to avoid adding bias to the estimate of $\mathbb{E}[X]$. In many domains constructing Y may prove to be difficult, while in others it is straightforward.

This method can be extended to the case of multiple antitheses $(Y^1, \ldots, Y^m)$ by averaging of the simulation and all of the m estimates as

$$W^m = \frac{X + Y^1 + \cdots + Y^m}{m+1} \tag{9}$$

3.1.4 Advantage Sum Estimators

In sequential decision-making settings, Zinkevich et al. [16] formulated a general approach to constructing a control variate called an *advantage sum estimator*. Advantage sum estimators utilize a value function defined over positions in the game to estimate the value of each decision made by the agents, as well as each decision made by chance. Advantage sum estimators split out the effects of luck (L_V), skill (S_V) and the positional advantage (P_V) when evaluating the decisions of agents. The luck, skill and positional advantage terms are described mathematically [13] as

$$S_{V_j}(z) = \sum_{\substack{ha \sqsubseteq z \\ P(h) \neq c}} V_j(ha) - V_j(h)$$

$$L_{V_j}(z) = \sum_{\substack{ha \sqsubseteq z \\ P(h) = c}} V_j(ha) - V_j(h)$$

$$P_{V_j} = V_j(\emptyset)$$

where ha refers to history h followed by action a. V_j represents the value function for player j which maps histories to real numbers. The function $P(h)$ determines which player is to act after history h. If $P(h) = c$, chance determines the action after history h. Then, for a given terminal history z and player j, the advantage sum estimator given by

$$\hat{\mu}_{V_j}(z) = S_{V_j}(z) + P_{V_j} \tag{10}$$

is an unbiased estimate of

$$\mu_j(z) = S_{V_j}(z) + L_{V_j}(z) + P_{V_j} \tag{11}$$

These estimators are provably unbiased when $\mathbb{E}[L_{V_j}(z)|\sigma]$ is zero, or in other words, when the value function, V, satisfies the zero-luck constraint.

3.2 Implemented Approaches

A number of variance reduction techniques have been successfully implemented for agent evaluation. We provide a brief summary of several of these implementations and their potential drawbacks, further motivating the need for simple and effective approaches to variance reduction.

3.2.1 Duplicate

In bridge [15] and computer poker [1] the notion of *duplicate* has seen success as an implemented antithetic variate method of reducing variance. The basic idea in these approaches is to generate a single deal of the cards, and then use each rotation of the player through the different sets of private cards as antithetic samples. Duplicate has been used by the Annual Computer Poker Competition to reduce the number of games that is necessary to run to determine a statistically significant winner. As previously stated in Section 2, two main drawbacks of using duplicate are its inability to measure individual skill in human play and the complexity of running duplicate matches as the number of agents increases.

3.2.2 Blackjack

In blackjack, Wolfe used the advantage sum estimator approach in a single-agent domain by utilizing a pre-computed strategy to compare the difference in the expected winnings of a player employing a fixed strategy to the expected winnings of a pre-computed near optimal strategy. [14]. At each decision point in the game, if the player's strategy differs from the pre-computed strategy, the player is credited the difference of the expected winnings. This adjustment is done under the assumption that the player will play the same as the pre-computed strategy from the point of deviation until the end of the game. Although this approach gives much lower per-hand variance, it is necessary to know beforehand, or be able to quickly compute, the expected value of each decision point in the game for the pre-computed strategy.

3.2.3 DIVAT

Within the domain of poker, a class of programs known as value assessment tools are also examples of advantage sum estimators. One such instance of these tools is DIVAT [2], a program that has been shown to reduce the variance in the two player limit poker variant. Much like the system used in blackjack, DIVAT compares decisions made in the actual games to that of a pre-computed strategy. Unlike blackjack however, the strategy being used is not following an optimal or near-optimal policy. Arguably the notion of an "optimal" policy, being one that maximizes the expected utility from each decision point, may be ill-formed in games such as poker [2]. DIVAT instead implements the advantage sum estimator idea, using a hand crafted value function for use in evaluation. The issue with DIVAT is that the creation of the value function it uses required a large amount of domain knowledge specific to a single variant of poker, rendering it ineffectual for use in even other variants of poker.

3.2.4 MIVAT

MIVAT is another value assessment tool that solves one shortcomings of DIVAT by learning a value function as opposed to hand crafting one [13]. Learning a value function allows for a much more general approach to creating instances of advantage sum estimators by subverting the need to have extensive amounts domain knowledge when creating the value function. White and Bowling showed that the learned value functions can obtain similar magnitudes of variance reduction as was seen with DIVAT in two player limit poker. Additionally, MIVAT was able to produce value functions used to reduce the variance in both six player limit and two player no-limit poker games. MIVAT does however suffer from the classic machine learning issues of feature construction and/or selection, which one might argue is still a form of domain knowledge.

3.2.5 Variance Reduction in Decision Making

Variance reduction techniques have also been employed in situations that involve decision making as opposed to agent evaluation. Two examples of this are the methods used in the Backgammon program TD_GAMMON [8] [9] and those explored by Veness and colleagues for use in Monte-Carlo Tree Search [12]. In TD_GAMMON, Tesauro used offline Monte-Carlo rollouts based on sample dice rolls to evaluate the decisions of the agent during the learning process. One potential pitfall of this approach is the cost associated with performing each of these rollouts as it may take thousands of samples in order to get a fairly confident estimate of the "best" move for a given position.

Veness et al. considered the use of both control variates and antithetic variates for use with the UCT Monte-Carlo Tree Search algorithm [5] [12]. The authors showed that both techniques are effective ways of reducing variance in the three test domains chosen to evaluate performance. Both of these techniques however suffered from difficulties related to implementation details. Building the antithetic estimator proved difficult in one of the domains, requiring a heuristic to generate the antithetic paths and failed in another domain since there were no obvious symmetries to exploit. Although the control variate estimators performed the best on the domains, they suffered from the need to have domain specific knowledge in order to create meaningful value functions.

Problems surrounding implementation, coupled with the others issues addressed in this section, show that eliminating the difficulty in creating effective low variance estimators is crucial in the practical deployment of any variance reduction technique.

4. BASELINE CONTROL VARIATES

We now introduce the *baseline* method of constructing control variates for agent evaluation in zero-sum games. Baseline allows for the creation of control variates in a simple and efficient manner, and is especially designed for settings with a large number of agents or instances of human play, where duplicate methods are either impossible or impractical. The baseline method only requires that a computer agent exists for the domain in question and that the random events observed in the simulation are reproducible.

The basic idea behind baseline is to compare the observed performance of the agents being evaluated to the performance of another computer agent in self-play. In this section we will describe this idea precisely and proceed to demonstrate experimentally that baseline can out perform duplicate in terms of reduction of the standard error. We also demonstrate that baseline is suitable for use in a wider range of domains such as human-machine competitions and Trading Agent Competition Ad Auction (TAC-AA) tournaments.

4.1 The baseline approach

Baseline is used to estimate the performance of an agent in a zero-sum domain where the stochasticity of that domain is captured in a random variable, denoted as Q. Essentially, this is any domain where the random seed can be controlled to produce exact replications of the chance events supplied by the environment. One example of such a domain would be the game of poker, where the randomness is based on the shuffle of the cards. Let $X = g(Q)$ be the random variable whose expectation $\mathbb{E}[X]$ is being estimated, where $g : Q \mapsto \mathbb{R}$ is a function specifying the performance the agent on instances of the domain. Let $(X_1, \ldots, X_n)$ be n samples from X corresponding to the measured performance of the agent on n samples from the domain Q: $(Q_1, \ldots, Q_n)$.

To apply the baseline method we assume that we have access to a computer agent (the control agent) for the domain in question, whose performance in self-play we will represent by another performance function $s : Q \mapsto \mathbb{R}$. The first step in the baseline approach is to create, for each domain sample Q_i, the control variate values $Y_i = s(Q_i)$. This is done by playing the control agent against itself on the instance of the domain described by Q_i and observing the utility gained. The *baseline estimator* $\bar{\beta}$ is then defined to be the control variate enhanced estimator obtained from using the control variate Y to estimate $\mathbb{E}[X]$, defined as follows.

$$\bar{\beta} = \frac{1}{n}\sum_{i=1}^{n} X_i + c\,(Y_i - \mathbb{E}[Y]) \tag{12}$$

Note that in general this does not eliminate the necessity of computing the expected value $\mathbb{E}[Y]$ of the control variate. One of the key ideas to the baseline approach is that as long as the game being played is zero-sum and any sample from Q assigns an equal probability for every possible assignment of agents to positions in the game, the expected value, $\mathbb{E}[Y]$, is 0. Combining this fact with the optimal coefficient from Equation 4 results in the final formula for the baseline estimator to be

$$\bar{\beta} = \frac{1}{n}\sum_{i=1}^{n} X_i - \frac{\widehat{\mathbb{C}\mathrm{ov}}[X, Y^{\sigma}]}{\widehat{\mathbb{V}\mathrm{ar}}[X]}(Y_i) \tag{13}$$

where $\widehat{\mathbb{C}\mathrm{ov}}[X, Y^{\sigma}]$ and $\widehat{\mathbb{V}\mathrm{ar}}[X]$ are the sample covariance and variance terms calculated from the observed samples. Using the sample variance and covariance terms introduces bias, however the estimator is still consistent [6], and the resulting reduction in variance can make it worth the trade off. For practical reasons, in cases where the control agent is making decisions in a stochastic manner the values for Y_i can be calculated using the average of multiple samples of the control agent's utilities on domain Q_i. If n is the number of samples being average, Y_i simply becomes

$$Y_i = \frac{1}{n}\sum_{j=1}^{n} s(Q_i) \tag{14}$$

In games with more than two agents, the baseline approach eliminates the complexity involved with creating duplicate matches. Furthermore, the baseline estimator is useful in domains where variance reduction is typically not considered due to implementation complexity such as human play evaluation. The simplicity of baseline provides a way of generating multiple estimators without having to explicitly encode any additional domain knowledge into the analysis. These properties are what make the baseline method an enticing approach to variance reduction in any appropriate domain where there exists even a single computer agent for use as the control.

5. EVALUATION METHODOLOGY

In Section 6 the baseline approach is experimentally evaluated on several data sets. In this section we describe the data sets and how the baseline approach was applied in each of these domains. Since most of these data sets are from the domain of Texas hold 'em poker, we begin with a brief explanation of this game.

5.1 Texas Hold 'em

Texas hold 'em is a variant of poker played using the standard 52 card deck. Although the game can be played with two or more players, the majority of the poker specific analysis performed has been performed on either the three player or two player (also known as heads up) versions of the game.

The gameplay of Texas hold 'em consists of four rounds, each of which involves cards first being dealt and then players making actions until the round terminates. The actions a player may choose are to either match the opponents total amount of money in the pot, raise the amount of money all other players must commit to the pot, or forfeit any money they currently have committed to the pot by folding their current hand.

The deal of the cards for the first round consists of two private face down cards for each player. At the start of the the second round three shared, or public cards, are dealt face up on the table. The next two rounds each have one more public card dealt out. After the four rounds are over the player with the best five card hand wins all of the money in the pot. This process is repeated many times, with the overall goal of a player being to maximizing their total winnings.

Texas hold 'em is played with many different betting structures, however only the no-limit or the limit betting structures are used in the analysis. No-limit betting means that any time a player can bet or raise, they may choose to bet any amount between a fixed minimum and the total amount of chip remaining to them. The limit betting structure fixes the size of each bet, typically being small in the early rounds and large in the later rounds of each hand.

5.2 Annual Computer Poker Competition

The Annual Computer Poker Competition (ACPC) is a yearly competition where academics and hobbyists submit poker playing computer agents to one of three Texas hold 'em competitions: heads up limit, heads up no-limit and three player limit. Hundreds of millions of duplicate poker hands are played for each of the competitions in order to separate out the competitors and determine a winner with statistical confidence.

The ACPC results from the 2011 three player limit and heads up no-limit competitions were analyzed using the baseline approach and compared to duplicate. Several different computer agents were used as the control agent in order to investigate the robustness of the baseline method. Each control agent participated in self-play on the exact same hands as the competition by using the same random seed and card dealing program. As the control agent is itself stochastic, to obtain a lower variance estimate of the control agent's performance on a given hand we had the agent play each hand fifty times, and averaged the performance over those fifty samples to obtain the final control agent performance value. Although each match of the competition is traditionally executed over millions of hands, we only utilized 180,000 hands from each of these matches in our experiments.

In order to compare the baseline approach to duplicate, baseline uses the same hands (two for every arrangement of cards) as the duplicate match, however, each pair of two hands are treated independently. For example, in the two-player case, let Q_j and Q'_j be the duplicates of the j-th hand, differing only in which player is assigned which private cards. Duplicate treats this as one hand-pair and computes the agent's performance on that hand as the average of the utility on the two separate hands $\frac{g(Q_j)+g(Q'_j)}{2}$, while baseline treats them as two separate hands with values $g(Q_j)-s(Q_j)$ and $g(Q'_j)-s(Q'_j)$, computing their respective control values independently.

In three-player Texas hold 'em, duplicate involves averaging over six different related hands, which correspond to all six different ways the three players can be assigned to the different sets private cards for a given hand. Baseline again treats these six separate hands separately and computes the control agent value independently for each. The results of these experiments are provided in Tables 1 and 2 and the post-play analysis is described in Section 6.1.

5.3 Man versus Machine Poker

The baseline approach was also evaluated on data gathered from two man versus machine (MvM) poker matches. In each of these matches, a team of two highly skilled professional poker players faced off against the same computer agent but on opposite sides of the cards. This was done to facilitate the computation of a duplicate score for the team. For reasons previously discussed, duplicate can only be used to evaluate a pair (team) of humans, but does nothing to reduce the variance in the estimation of a single human's performance against an opponent. In contrast, baseline allows us to obtain lower variance analysis on the individual performance of each of the two human players. Once again, in order to compare baseline to duplicate, the hands for the two human players were treated as both duplicate and independent. Due to the high cost of gathering human data of this nature, only 1000 hands were played in each of the matches. Identical to the ACPC analysis, the average of fifty samples of the control policy in self play on each hand was used when computing the baseline control values. Post-play analysis of two man versus machine matches is presented in Section 6.2 and the results are shown in Tables 3 and 4.

5.4 Trading Agent Competition

Lastly, the baseline method was tested on a mock Ad Auction tournament from the Trading Agent Competition (TAC-AA) [11]. For this experiment, six ad auction agents were gathered from the TAC repository and faced off in a eight player tournament, with two random agents filling the empty positions. The tournament consisted of 100 matches, where each match was a simulation of ten days, after which the payouts were converted such that the resulting game was zero-sum. For each of the matches the agents are each assigned a random specialization. The specializations serve as a way for an agents to capitalize on the simulated marketplace. We used the specializations as a way to capture the environmental stochasticity, such that the exact same assignment of specializations seen on each day in the tournament match those used in the control agent's self play. A subset of the competition agents were chosen as the different control agents for the various baseline estimators. For practicality reasons, each self play match was only sampled

one time for this domain. The performance of the different baseline estimators for this tournament are shown in Table 5 and discussed in Section 6.3

6. RESULTS

In this section the results for the experimental evaluation of the baseline approach are presented. For each of the poker related experiments, the utility for each agent is reported in milli-big blinds per hand (mbb/h), which is a standard unit of measurement within the domain of poker. For the TAC-AA results, the utility of the agents is reported in thousands of dollars.

Each of the rows in a table refers to a specific agent being evaluated, either in its performance against all other agents (if no other agent is listed) or in its performance against one specified opponent. The $\bar{X}$ column in each of the tables shows the mean of the raw utility as well as the size of the 95% confidence interval, which was computed as $\frac{1.96\hat{\sigma}}{\sqrt{n-1}}$ where n is the number of samples and $\hat{\sigma}$ is the sample standard deviation over those n samples. The remaining columns of each table are labeled with the name of the estimator being used to estimate the performance of the agents. The "Dup" label refers to the duplicate estimator and a $\bar{\beta}$ label refers to a baseline estimator.

The values in each of the estimator columns show the reduction of the sample standard error for the specified estimator over that of $\bar{X}$. The reduction values are displayed as a percentage and the sample standard error for each of the estimators was computed as $\frac{\sigma}{\sqrt{n}}$. A value of 0% would mean that there is no difference between the size of the standard error between the estimator and simply using the raw utility. Conversely, a value of 100% would mean that the estimator was able to reduce the standard error to 0. Simply put, estimators with large percentage reduction values perform better than ones with small percentage reduction values. Finally, a negative percentage reduction value indicates that the standard error actually increased when using the specified estimator.

6.1 ACPC

We begin with the results of the baseline approach applied to the data from the 2011 Annual Computer Poker Competition.

6.1.1 Three Player Limit

The summary of the performance estimate for each of the competitors in the three player limit competition is shown in Table 1. This table shows that the baseline estimator outperforms the duplicate counterpart for all nine agent summaries. Also, the baseline estimator never performs worse than naively estimating the expected value simply on raw utility. This is not the case for duplicate, which actually increased the size of the sample standard error in over half of the summaries. One possible explanation for this could be that the full six-way duplicate is not the optimal way to combine the possible player orderings

In addition to its lower performance, using the full player ordering to create a three player duplicate match increases the logistical complexity. While being feasible approach for computer competitions, having six players on a team to represent just one person could prove to be very difficult when used for human evaluation. Baseline provides a much simpler framework, requiring only one such ordering and just three players. It is clear from these results that baseline should be utilized for competitions of this nature.

Match	$\bar{X}$	Dup (%)	$\bar{\beta}$ (%)
Bnold3	-10.56 ± 1.10	8.04	**37.20**
dcubot3plr	6.16 ± 2.03	39.52	**49.68**
Entropy	-1.64 ± 1.67	-40.41	**22.57**
Hyperborean-iro	20.64 ± 0.98	-0.61	**39.08**
Hyperborean-tbr	16.74 ± 1.07	-3.03	**33.28**
LittleRock	10.25 ± 0.91	-6.32	**35.56**
OwnBot	-0.4 ± 2.68	33.07	**41.95**
player_zeta_3p	-51.96 ± 2.50	-25.54	**21.78**
Sartre3p	23.01 ± 1.38	28.12	**46.20**

Table 1: ACPC 3P Limit (180K hands)

6.1.2 Heads Up No-Limit

Table 2 shows the results of the estimators when applied to the data from the heads up no-limit ACPC tournament data. The columns labeled $\bar{\beta}_1$, $\bar{\beta}_2$ and $\bar{\beta}_3$ refer to the baseline estimators resulting by using three different control agents and the final $\bar{M}$ column refers to the multiple control variate situation with β_1, β_2 and β_3 being the columns of the $\vec{Y}$ vector in Equation 6.

One observation that can be made from this data is the although duplicate provided estimates with the lowest sample standard error, there were not many matches where the magnitude of this reduction was significantly larger than any of the baseline approaches. Also noticeable from this table is the lack of separation between each of the baseline approaches. For almost every match, the baseline reduction percentages are nearly identical to one another, which explains why the $\bar{M}$ estimator is unable to gain much in the way of additional reduction over any individual baseline. For this competition, baseline shows a competitive method for variance reduction and should be considered for competitions of this nature when producing duplicate results is not as straight forward as it is in the game of poker.

6.2 MvM Poker

Tables 3 and 4 summarize the results of the two man versus machine matches. In both of the matches, the results show that the baseline estimator is able to reduce the sample standard error for both the individual and team results. In the case of the team data, although duplicate slightly outperforms baseline in the first match, both of the estimators end up producing statistically significant results. Duplicate and baseline perform equally well on the team data in the second match. Baseline in this setting has a huge advantage over duplicate since it can be used to estimate each of the players's individual play rather than just the team data.

The need for splitting out the individual players is reinforced in the analysis of the two matches. It is clear that one of the human players, denoted as Human 1 in the tables, was the better player in the first match. The second human player, denoted as Human 2 in the tables, actually lost to

Match	$\bar{X}$	Dup (%)	$\bar{\beta}_1$ (%)	$\bar{\beta}_2$ (%)	$\bar{\beta}_3$ (%)	$\bar{M}$ (%)
0 vs 1	-333.87 ± 294.25	**2.03**	1.13	1.54	1.45	1.58
0 vs 2	-2841.77 ± 136.99	**2.62**	0.95	1.18	1.09	1.21
0 vs 3	-1318.08 ± 241.95	**3.82**	1.22	1.61	1.41	1.64
0 vs 4	17785.11 ± 480.36	**32.55**	4.62	8.01	6.92	8.12
0 vs 5	-1835.21 ± 123.15	**2.16**	0.56	0.63	0.60	0.67
0 vs 6	-4954.7 ± 209.86	**7.97**	1.31	1.66	1.58	1.72
1 vs 2	-592.08 ± 81.31	**9.95**	6.78	9.26	8.50	9.45
1 vs 3	734.91 ± 65.43	**4.45**	3.15	4.26	3.71	4.32
1 vs 4	4586.91 ± 79.18	**32.29**	7.54	12.71	10.93	12.83
1 vs 5	731.65 ± 51.78	**8.01**	3.46	4.72	3.97	4.78
1 vs 6	-280.03 ± 74.52	9.32	5.97	9.29	8.05	**9.34**
2 vs 3	561.89 ± 73.40	**9.31**	6.19	8.51	7.67	8.65
2 vs 4	2195.48 ± 101.12	**11.66**	3.53	5.58	5.07	5.69
2 vs 5	593.43 ± 63.02	**9.96**	6.50	8.68	7.84	8.85
2 vs 6	112.17 ± 69.48	**20.85**	13.97	19.08	18.60	19.94
3 vs 4	9877.91 ± 208.98	**7.98**	4.10	6.66	5.71	6.70
3 vs 5	-95.78 ± 73.47	**13.71**	3.46	4.75	3.96	4.80
3 vs 6	-434.35 ± 79.64	**11.90**	5.88	9.51	7.63	9.51
4 vs 5	-2595.83 ± 58.87	**11.13**	5.40	8.45	7.23	8.48
4 vs 6	-1838.68 ± 52.04	**6.04**	3.59	5.14	4.60	5.20
5 vs 6	-330.09 ± 72.26	**10.37**	6.21	8.77	7.82	8.88

Table 2: ACPC Heads Up No-limit (180K hands)

Match	$\bar{X}$	$\bar{\beta}$ (%)	Dup (%)
Human 1	39.55 ± 962.60	**17.84**	—
Human 2	-1597.75 ± 1340.45	**18.54**	—
Team	-779.1 ± 825.71	17.66	**25.1**

Table 3: MvM Match 1 (1K hands)

Match	$\bar{X}$	$\bar{\beta}$ (%)	Dup (%)
Human 1	78.78 ± 912.61	**18.40**	—
Human 2	-146.63 ± 866.92	**34.89**	—
Team	-33.93 ± 629.23	**25.40**	24.39

Table 4: MvM Match 2 (1K hands)

the computer agent with statistical significance, even when using just raw utility as the estimate. Both baseline and duplicate show that in the first match, the human team also lost with significance. This shows that the loss observed by the team in the first match was due to the gap in the skill between Human 1 and Human 2. By pairing them as a team in the duplicate setting, Human 1 unfairly loses the first match, instead of it being a result of his/her own play. These results demonstrate the advantage of using baseline as the estimator, since individual play is a much more appealing competition for human-machine competitions.

6.3 TAC-AA

The results of using baseline in a mock Ad Auction TAC tournament are shown in Table 5. In this experiment, each of the baseline estimators was able to reduce the size of the sample standard error for all of the matches and the resulting reduction percentages range from 0.1% up to 53%. For this tournament, the baseline estimator $\bar{\beta}_3$ provided the greatest reduction in the standard error out of all the baseline estimators used. Notably, none of the baseline estimators ever increase the standard error. Given that the tournament consisted of six competition agents and two random agents, it is unclear what impact this had on the performance of the baseline estimator. It is possible that in a tournament using eight competition agents, the baseline estimator may

Player	$\bar{X}$	$\bar{\beta}_1$ (%)	$\bar{\beta}_2$ (%)	$\bar{\beta}_3$ (%)
tau	1.9 ± 1.35	7.78	7.64	**19.98**
crocodile11	6.69 ± 1.69	1.36	1.28	**5.82**
crocodile	8.39 ± 1.98	18.30	5.65	**20.31**
epfl	9.42 ± 2.44	2.26	2.80	**5.34**
metroclick	10.19 ± 2.84	9.72	12.59	**53.25**
Schlemazl	9.14 ± 1.81	0.12	0.01	**0.17**

Table 5: Mock TAC-AA Tournament (100 matches)

provide larger gains. Additionally, only a single sample of the baseline estimator was used when calculating the baseline scores, which could have potentially added noise into the estimate. Regardless, this experiment not only demonstrates that baseline estimators can be easily produced from a variety of agents, but the resulting estimators can achieve a significant reduction in the variance of the tournament. Baseline should be definitely be used in domains that exhibit the properties present in TAC-AA type tournaments.

7. CONCLUSIONS

The baseline approach provides a simple way of creating control variates and can be used in any zero-sum domain with readily available agents. Baseline estimators have the advantage of not requiring prior domain knowledge or even a known expected value for the control, problems that can typically lead to forgoing the use of variance reduction techniques due to the overhead involved with implementation. We showed that these estimators are not only competitive with state-of-the-art methods in two player computer poker, but are superior to the best known techniques in the domain of three player limit computer poker.

Baseline estimators should be used in domains where variance reduction is not typically applied, such as those involving human players or in TAC-AA type tournaments. For instances of human play where the duplicate variance reduction technique can be applied, baseline estimators proved to be as good as the duplicate estimator. Furthermore, baseline estimators allow for individual analysis in human domains, something that duplicate cannot achieve.

The baseline approach is one that should be used in any applicable domain due to its simplicity and potential as a strong variance reduction method. In particular, baseline should replace duplicate in the ACPC three player limit matches.

8. REFERENCES

[1] Annual Computer Poker Competition. `http://www.computerpokercompetition.org`.

[2] D. Billings and M. Kan. A tool for the direct assessment of poker decisions. *The International Computer Games Association Journal,*, 29(3):119–142, 2006.

[3] P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, 1962.

[4] International Federation of Poker. `http://www.pokerfed.org/`.

[5] L. Kocsis and C. Szepesvári. Bandit based Monte-Carlo planning. In *ECML*, pages 282–293, 2006.

[6] B. L. Nelson. Control Variate Remedies. *Operations Research*, 38(6):974–992, Nov. 1990.

[7] S. M. Ross. *Simulation (2. ed.)*. Statistical modeling and decision science. Academic Press, 1997.

[8] G. Tesauro. Temporal difference learning and TD-Gammon. *Communications of the ACM*, 38(3):58–68, 1995.

[9] G. Tesauro. Programming backgammon using self-teaching neural nets. *Artificial Intelligence*, 134(1-2):181–199, Jan. 2002.

[10] The Second Man-Machine Poker Competition. `http://webdocs.cs.ualberta.ca/~games/poker/man-machine/Press/PR-05-10-2008/`.

[11] Trading Agent Competition Ad Auctions. `http://aa.tradingagents.org/`.

[12] J. Veness, M. Lanctot, and M. Bowling. Variance reduction in monte-carlo tree search. *Advances in Neural Information Processing Systems 24*, pages 1836–1844, 2011.

[13] M. White and M. Bowling. Learning a value analysis tool for agent evaluation. *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1976 – 1981, 2009.

[14] D. Wolfe. Distinguishing gamblers from investors at the blackjack table. In *Computers and Games 2002, LNCS 2883*, pages 1–10. Springer-Verlag, 2002.

[15] World Bridge Federation. `http://www.worldbridge.org/home.asp`.

[16] M. Zinkevich, M. Bowling, and N. Bard. Optimal unbiased estimators for evaluating agent performance. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI)*, pages 573–578, 2006.

APPENDIX

A. ACPC NO-LIMIT AGENT NAMES

Identifier	Full Name	Identifier	Full Name
0	player_kappa_nl	4	POMPEIA
1	hugh	5	Rembrant
2	Hyperborean-iro	6	SartreNL
3	Lucky7		

Table 6: ACPC No-Limit Agent Names