The Knowledge Gradient for Optimal Learning

Warren B. Powell

May 2, 2010

Abstract

Optimal learning addresses the problem of how to collect information so that it benefits future decisions. For off-line problems, we have to make a series of measurements or observations before choosing a final design or set of parameters; for online problems, we learn from rewards we are receiving, and we want to strike a balance between rewards earned now and better decisions in the future. This chapter reviews these problems, describes optimal and heuristic policies, and shows how to compare competing policies. Then, the presentation focuses on the concept of the knowledge gradient, which guides information collection by maximizing the marginal value of information. We show how this idea can be applied to both online and off-line problems, as well as a broad range of other applications which have not previously yielded to formal techniques.

1 Introduction

There is a wide range of problems where we need to make a decision under some sort of uncertainty, and where we have the ability to collect information in some way to reduce this uncertainty. The problem is that collecting information can be time consuming and/or expensive, so we have to do this in an intelligent way. Some examples include

- We wish to find a supplier for a component who provides the lowest cost and best level of service. We know the cost, but the only way to evaluate the level of service is to try the supplier and observe actual delivery times.
- We need to find the best path through New York City to respond to certain emergencies (e.g. getting a fire truck to city hall). To evaluate the time required to traverse each path, we can try each path, or we could collect information about specific links in the network. Which paths (or links) should we measure?
- We are trying to find the best molecular compound to produce a particular result (curing cancer, storing energy, conducting electricity). There are thousands of combinations we can try, but each test takes a day. Which compounds should we test?
- We would like to find the best price to sell a product on the internet. We can experiment with different prices and measure sales. How do we go about testing different prices?
- We have a simulation model of an operational problem (for example, managing takeoffs and landings at Heathrow Airport). We have a number of control parameters that we can adjust to improve performance, but running a simulation takes half a day. How do we go about trying out different parameters?
- We are running a stochastic search algorithm to solve a specific optimization problem. Each iteration takes several minutes to perform a noisy function evaluation (this might be a simulation of the climate). How do we optimize this function as quickly as possible?

These are just a small number of examples of problems where we face the challenge of collecting information to make better decisions. Applications range from business decisions, science and engineering, and simulation and stochastic optimization.

The learning problems that we wish to address all have three fundamental components: a measurement decision, which determines what information we are going to collect; the information that we obtain; and an implementation decision which uses the information. We are primarily interested in sequential problems, where we can make multiple measurements and where each measurement may depend on the outcomes of previous measurements. Our goal, then, is to design a *measurement policy* which determines how these measurements are made.

Measurement problems come in two fundamental flavors. Off-line problems use a sequence of measurements before making a final implementation decision. These arise when we have a period of time to do research before choosing a final design. The second flavor is on-line problems, where we collect information each time we make an implementation decision. For example, we may want to find the best path from our new apartment in New York City to our new job. The only way to evaluate a path is to try it. The implementation decision is the choice of path to use tomorrow, but we collect information as we try the path. Off-line problems are often grouped under labels

such as ranking and selection (which generally refers to choosing among a finite set of alternatives), stochastic search, and simulation-optimization. On-line problems are usually referred to as bandit problems. Our presentation of the knowledge gradient treats both off-line and on-line problems in an integrated way.

Information collection problems can be approached from two perspectives: frequentist and Bayesian. In the frequentist approach, we form estimates about a truth based on information drawn from a sample (see **?**, **?**, **?** and **?**) for excellent surveys of learning from a frequentist perspective). In this article, we focus on the Bayesian perspective where we assume that we have a prior belief about problem parameters, along with estimates of the level of uncertainty. The Bayesian perspective assumes that there is a prior distribution of possible truths, and our goal is to design a policy that discovers this truth as quickly as possible. The frequentist perspective uses noisy observations from an unknown truth to make statistical statements about this truth; the Bayesian perspective uses a prior belief and noisy measurements to create a probability distribution to describe the truth.

In this short article, we adopt a Bayesian perspective, focusing on a concept we call the *knowledge gradient* which belongs to a family of methods which guides the learning process based on the marginal value of information. The roots of this idea in the field of decision theory belong to the seminal paper by **?** on the value of information. We first became aware of this idea applied as a policy for off-line ranking and selection problems from **?**. A separate line of research which uses arose in the context of optimizing unknown functions under the general name "global optimization," with roots in the seminal paper by **?**. When these functions can only be measured with uncertainty, this field has been referred to as Bayesian global optimization, with many contributions (see **?** for a nice review). Our work in this area is based on **?** which analyzes the properties of sequential measurement policies based on the marginal value of a single measurement using a Bayesian framework with variance known; **?** provides an important generalization to problems with correlated beliefs (reviewed below). In parallel research, **?** derive the marginal value of information for the important case where the variance is unknown using the name $LL(1)$ (linear loss, with measurement batches of one observation).

We provide a brief summary of different policies for collecting information. Although measurement policies may have a Bayesian or frequentist foundation, it is possible (and we would argue most natural) to evaluate any policy by sampling a truth from a prior distribution, and then determine how well a policy learns this assumed truth. This exercise then has to be repeated over many possible truths drawn from the prior.

The essential feature that separates learning problems from traditional stochastic optimization is that we are unsure about our uncertainty. If we make an observation, we are willing to update the probability distributions we use to describe uncertain parameters. Inserting this step of updating our beliefs after a measurement is made is typically referred to as statistical learning. When we actively make choices of what to measure, taking into account our willingness to update our beliefs, then this is optimal learning.

This chapter is intended to serve as a brief introduction to some of the important problem classes in optimal learning. We provide a mathematical framework for formulating and evaluating measurement policies. We review a number of heuristic policies, along with optimal policies for special problem classes. The remainder of the chapter focuses on the concept of the knowledge gradient algorithm which is a measurement policy that applies to a wide range of applications.

2 Elementary problems

There are two elementary learning problems that provide the foundation for our discussion. Both have a discrete set of measurements $\mathcal{X} = (1, 2, \ldots, M)$ where M is an integer which is "not too large" which means we do not have any difficulty enumerating the choices. The first problem is known in the literature as the multiarmed-bandit problem, which is an on-line learning problem where we learn from the rewards that we receive. The second is known as the ranking and selection problem, where we have a budget of N measurements to evaluate each choice, after which we have to decide which alternative appears to be best.

2.1 Multiarmed bandit problems

The multiarmed bandit problem is based on the story of trying to choose the best of a set of M slot machines (often known as one-armed bandits). We do not know how much we will win each time we play a particular slot machine, but we have a distribution of belief, which we acknowledge may be wrong. We may think one machine has the highest expected reward, but we are willing to acknowledge that we may be wrong and another machine may also be the best. The only way we will learn is to try machines that do not appear to be the best. But while trying these machines, we may be earning lower rewards than we would earn by playing the machines that we think are better. The goal is to maximize the expected discounted sum of rewards that balance what we earn against what we learn (to improve future decisions).

Let μ_x be the true mean reward if we choose x. We do not know μ_x, but assume that we believe that it is normally distributed with prior mean μ_x^0 and variance $(\sigma_x^0)^2$. For convenience, we define the precision $\beta_x^0 = 1/(\sigma_x^0)^2$. Let W^n be the reward ("winnings") we receive in the nth iteration, and let (μ^n, β^n) be our vector of beliefs about the means and precisions for all the choices after n measurements. We can write $\mu_x^n = \mathbb{E}^n \mu_x$, where $\mathbb{E}^n$ is the expectation given the first n measurements. Let $S^n = (\mu^n, \beta^n)$ be our "state of knowledge" (often referred to as the belief state).

We let x^n be the choice we make after n measurements, meaning that our first choice is x^0 which is based purely on the prior. We make these measurements using a policy π which is allowed to depend on the history of observations $W^1, W^2, \ldots, W^n$. Let $X^{\pi,n}(S^n)$ be the random variable representing the decision we make, given our state S^n, and given measurement policy π. This notation allows our policy to depend on n; if we wish to follow a stationary policy, we would write it as $X^{\pi,n}(S^n)$ (in a finite-horizon problem, the policy can depend on the number of measurements n, as well as depending on the belief state S^n). Our goal is to find a measurement policy π that solves

$$\sup_{\pi} F^{\pi} = \mathbb{E}^{\pi} \sum_{n=0}^{N} \gamma^n \mu_{X^{\pi,n}(S^n)} \tag{1}$$

where γ is a discount factor. We write the expectation $\mathbb{E}^{\pi}$ as dependent on the policy π, which reflects assumptions on how we construct the underlying probability space. We assume that an elementary outcome is a sequence of decisions of what to measure (which depend on the policy) and the results of a particular measurement. This is not the only way to construct the probability space, but it is the one that is most often used in the research community.

In the classical multiarmed bandit problem, $N = \infty$ and $\gamma < 1$, but the finite horizon problem

(possibly with $\gamma = 1$) is also of interest.

2.2 Ranking and selection problems

Now imagine that we have a budget of N measurements (or B dollars to spend on measurements) after which we have to choose the best of a set of M alternatives. In the bandit problem, we learn as we go, incurring rewards (or costs) as we proceed. In the ranking and selection problem, we are not concerned with how well our choices perform during the process of collecting information. Instead, we are only concerned with how well our final choice performs.

Let μ_x^N be the posterior mean of the value of alternative x after N measurements, which we chose using measurement policy π. μ_x^N is a random variable whose distribution depends on our measurement policy. The value of using π can be written

$$F^\pi = \mathbb{E} \max_x \left[\mu_x^N\right] = \mathbb{E}^\pi \mu_{x^\pi}^N.$$

where $x^\pi = \arg\max_x \mu_x^N$ is the optimal solution when we follow measurement policy π. The problem of finding the best policy is then given by

$$\sup_{\pi \in \Pi} F^\pi.$$

We note that it is possible to write the objective as

$$\sup_\pi \mathbb{E} F^\pi$$

in which case we assume that the measurement policy is built into the objective function. When we write the objective this way, it means that the probability space consists of all potential measurements W_x^n for all alternatives x and all measurements n. Alternatively, we can write the objective as

$$\sup_\pi \mathbb{E}^\pi F.$$

Written this way, it means that we have imbedded the policy into the probability space (a sample outcome consists of measurement alternatives and realizations) which means that F does not explicitly depend on the policy.

2.3 Notes on objective functions

The formulations we have given in this section assume that we are maximizing a reward. Equivalently we could minimize the expected opportunity cost. For the ranking and selection problem, this would be written

$$EOC = \mathbb{E} \max_x \mu_x - \mathbb{E} \mu_{x^\pi}$$

where we are measuring the value that we could achieve if we could find the best alternative using the true means, versus the value of the alternative we did choose (but again using the true means). Keep in mind that the expectations here are over both the distribution of truths as well as the distribution over measurements. Minimizing the expected opportunity cost is the same as solving

$$\sup_{\pi} \mathbb{E}\mu_{x^\pi}.$$

It is also possible to show that

$$\sup_{\pi}\sup_{\chi} \mathbb{E}^\pi \mu_{\chi(x)} = \sup_{\pi} \mathbb{E}^\pi \max_x \mu_x^N, \tag{2}$$

where χ is the policy for choosing the best alternative x given what we measure, which for our problem is defined by

$$\chi = \arg\max_x \mu_x^N.$$

Equation (**??**) states that if we use a measurement policy to find the best alternative x^π and we evaluate this choice using the true values μ_x, we get the same answer if we evaluate our choice using the estimates μ_x^N.

3 Learning

At the heart of any learning problem is not only uncertainty about the value of the choices we are making, but also uncertainty about our uncertainty. Learning problems are easily posed in a Bayesian framework, where we are able to capture the uncertainty in our belief about a system. In our bandit or ranking and selection problems, μ_x is the true value of x, but we do not know this value. Instead, we assign a probability distribution that describes what we think μ_x is for each x. Before we start collecting any information, we might assume that our *prior* distribution of belief about μ_x is normally distributed with mean μ_x^0 and variance $(\sigma_x^0)^2$. We adopt a common convention in Bayesian analysis and define the *precision* of our belief as $\beta_x^0 = 1/(\sigma_x^2)^2$. Now assume that when we make an observation W_x^n of choice x, the precision of this measurement is known and given by β_ϵ. To reduce notational clutter, we assume that this is constant across x, but this is easily relaxed.

Assume that our current (prior) belief about choice x is given by μ_x^n and β_x^n, and that we choose to measure alternative $x^n = x$. We would then observe W_x^{n+1}, which is unknown when we picked x^n. Further assume that our prior is normally distributed (that is, $\mu_x \sim N(\mu_x^0, 1/\beta_x^0)$, and that $W_x^{n+1} = \mu_x + \varepsilon_x^{n+1}$, where ε_x^{n+1} is also normally distributed with precision $\beta_{epsilon}$. Bayes theorem can be used to show that our updated mean and precision of the posterior belief can be computed using

$$\mu_x^{n+1} = \frac{\beta_x^n \mu_x^n + \beta_\epsilon W_x^{n+1}}{\beta_x^n + \beta_\epsilon}, \tag{3}$$

$$\beta_x^{n+1} = \beta_x^n + \beta_\epsilon. \tag{4}$$

We also have the property that if our prior belief about μ is normally distributed, then the posterior belief is normally distributed.

4 Measurement policies

Our central challenge is designing a policy for collecting information. For off-line problems, these policies guide information collection that determines the final decision. For online problems, our policy has to strike a balance between receiving rewards (or incurring costs) and collecting information that will help future decisions.

Measurement policies can be deterministic or sequential. If we are using a deterministic policy, decisions about what to measure are made before any measurements are learned. Deterministic policies can be optimal for certain types of statistical learning problems. Our interest is in sequential problems, where the next measurement decision is made only after learning the value of the previous measurement.

We begin by showing that optimal measurement policies can be characterized using a simple dynamic programming formulation. The problem is that the dynamic program cannot be solved exactly. We then describe a series of simple heuristic policies that are often used in the research community.

4.1 Optimal sequential policies

Dynamic programming is widely used for sequential decision problems. For our problem, we start in a (knowledge) state S^n, then we take an action (measurement) x^n, observe a random outcome $W_{x^n}^{n+1}$, which takes us to a new state according to the transition function $S^{n+1} = S^M(S^n, x^n, W^{n+1})$ defined by equations (**??**) and (**??**). Our optimal measurement x^n can, in theory, be characterized using Bellman's equation:

$$V(S^n) = \max_x \big(C(S^n, x) + \gamma \mathbb{E}\left\{ V(S^{n+1}) \big| S^n \right\}\big). \tag{5}$$

Here, $V(S^n)$ captures the value of being in a particular knowledge state S^n. Using the principle of dynamic programming, this is given by choosing the measurement action x that maximizes the contribution earned (this might be a negative cost) from taking action x, plus the value of being in the knowledge state S^{n+1} that results from this action, given by $V(S^{n+1})$. Of course, S^{n+1} is random given S^n and x, so we have to take its expectation over the measurement W_x^{n+1} for each alternative x. For off-line problems, $C(S^n, x)$ is zero, since we do not receive any value until we pick the final design. For on-line problems, $C(S^n, x)$ would be the expected reward μ_x^n.

So, if equation (**??**) gives us the optimal measurement policy, why don't we just use this solution? The reason is that we just do not have algorithms to solve dynamic programs when the state variable is a vector of continuous parameters. There is, however, a special case that can be solved optimally without directly solving the dynamic program, which uses the concept of Gittins indices for multiarmed bandit problems. After presenting this idea, we review the most popular heuristic policies. Our presentation focuses purely on sequential policies where the choice of the *nth* measurement depends on the prior and the outcomes of $W^1, \ldots, W^n$. We then close by introducing the knowledge gradient policy and illustrate the wide range of information collection problems that can be addressed using this strategy.

4.2 Gittins indices for multiarmed bandit problems

In 1974, **?** found that the multiarmed bandit problem (specifically, the infinite horizon version of the multiarmed bandit problem) could be solved optimally using an index policy computed using

$$\Gamma(\mu_x^n, \sigma_x^n, \sigma_\epsilon, \gamma) = \mu_x^n + \Gamma\left(0, \frac{\sigma_x^n}{\sigma_\epsilon}, 1, \gamma\right)\sigma_\epsilon. \tag{6}$$

In this expression, $\Gamma(\mu_x^n, \sigma_x^n, \sigma_\epsilon, \gamma)$ is the Gittins index for measurement x for which the belief is μ_x^n and the standard deviation is σ_x^n. $\sigma_\epsilon = \sqrt{1/\beta_\epsilon}$ is the standard deviation of the measurement and γ is the discount factor. The Gittins index policy, as it became known in the literature, specifies that you measure the choice x with the highest value of $\Gamma(\mu_x^n, \sigma_x^n, \sigma_\epsilon, \gamma)$. The power of this policy is that there is one index per choice, and at no time are we dealing with multidimensional state vectors.

This leaves us with the challenge of computing $\Gamma(\mu_x^n, \sigma_x^n, \sigma_\epsilon, \gamma)$. Equation (**??**) shows us that we can compute the index as a function of a simpler index $\Gamma(0, \frac{\sigma_x^n}{\sigma_\epsilon}, 1, \gamma)$. This has a close parallel with the standard normal distribution (indeed, this relationship depends on normally distributed beliefs). While this relationship simplifies the problem, we still have to find $\Gamma(0, \frac{\sigma_x^n}{\sigma_\epsilon}, 1, \gamma)$, which requires the fairly difficult numerical solution of an integral equation.

Fortunately, there have been recent efforts to develop numerical approximations. **?** found that

$$\Gamma(0, s, 1, \gamma) = \sqrt{-\log\gamma}\; b\left(-\frac{s^2}{\log\gamma}\right) \tag{7}$$

where the function $b(\cdot)$ has to be approximated. **?** improved on the initial approximation of this function by **?**, proposing

$$b(\zeta) \approx \begin{cases} \frac{\zeta}{\sqrt{2}} & \zeta \leq \frac{1}{7} \\ \exp\left(-0.02645(\log\zeta)^2 + 0.89106\log\zeta - 0.4873\right) & \frac{1}{7} < \zeta \leq 100 \\ \sqrt{\zeta}\sqrt{\zeta\log\zeta - \log\log\zeta - \log 16\pi} & \zeta > 100. \end{cases} \tag{8}$$

The Gittins index policy is specifically designed for online problems, and in particular infinite horizon bandit problems. However, the structure of the policy offers some insights into the information collection process, as we see below.

4.3 Heuristic policies

There are a number of simple policies that can be used for collecting information. The policies reviewed below are the most popular. These can generally be adapted for online or off-line applications.

- Pure exploration - Pure exploration involves picking choices at random. If there are M choices, we might choose $x^n = x$ with probability $1/M$ or use some other exogenously driven process. A pure exploration strategy makes little sense in an online application, but it can be useful (if not optimal) for off-line problems, especially for high-dimensional applications where the measurement space $\mathcal{X}$ is quite large.

- Pure exploitation - We make the decision that appears to be best given what we know. Stated mathematically, a pure exploitation policy would be written

$$x^n = \arg\max_{x \in \mathcal{X}} \mu_x^n.$$

Pure exploitation can be effective for on-line learning where the prior information is quite good.

- Mixed exploration and exploitation - Here we explore with some probability ρ, and exploit with probability $1 - \rho$. A variation is epsilon-greedy exploration, where we explore with probability $\rho^n = c/n$ where c is a tunable parameter.

- Boltzmann exploration - Here we explore with probability

$$\rho_x^n = \frac{\exp(\theta \mu_x^n)}{\sum_{x' \in \mathcal{X}} \exp(\theta \mu_{x'}^n)},$$

where θ is a tunable parameter. $\theta = 0$ produces pure exploration, while as θ increases, it approaches pure exploitation. This policy explores in a more intelligent way than the more classical exploration policies.

- Interval estimation - Interval estimation is the most sophisticated of this group of heuristics. Here, we compute an index using

$$\nu_x^n = \mu_x^n + z_\alpha \bar{\sigma}_x^n. \tag{9}$$

In this policy, z_α is a tunable parameter, and $\bar{\sigma}_x^n$ is the standard deviation of our estimate μ_x^n. This policy strikes a balance between the estimate μ_x^n (low values are less likely to be tested), and the uncertainty with which we know μ_x^n. The policy rewards higher levels of uncertainty, but only if μ_x^n is competitive.

It is useful to compare the index policy given by interval estimation in equation (**??**) with the Gittins index policy, given in equation (**??**). Both policies compute an index given by the current belief plus an additional "bonus" (often referred to as the uncertainty bonus). How this bonus is computed is where the two policies differ. The Gittins policy uses a theoretically derived factor which declines to zero as the number of observations increase. This is multiplied by the standard deviation σ_ϵ of the measurement error. For interval estimation, the bonus is a constant factor (which has to be tuned) times the standard deviation $\bar{\sigma}_x^n$ of our estimate μ_x^n which then declines with the number of observations. As the number of times we measure an alternative increases, the factor $\Gamma(0, \frac{\sigma_x^n}{\sigma_\epsilon}, 1, \gamma)$ in equation (**??**) decreases (σ_ϵ stays the same). By contrast, with interval estimation, it is $\bar{\sigma}_x^n$ that decreases, while z_α stays the same.

4.4 Policies from simulation optimization

An entire body of research has developed around the problem of choosing the best set of parameters to guide a simulation. If these parameters are discrete, and if we fix the length of the simulation, this problem falls under the umbrella of ranking and selection if we are considering a finite number of alternatives. Simulation optimization introduces the additional dimension that we can choose the length of the simulation, but we may face a budget on total computing time. This problem was first

addressed under the name of "Optimal Computing Budget Allocation" (OCBA) by **?**. This idea has subsequently been studied in a number of papers (**???**, and **?**). **?** introduces the $LL(B)$ strategy which maximizes the linear loss with measurement budget B. **?** introduce an OCBA procedure for optimizing the expected value of a chosen design, using the Bonferroni inequality to approximate the objective function for a single stage. A common strategy in simulation is to test different parameters using the same set of random numbers to reduce the variance of the comparisons. **?** apply the OCBA concept to measurements using common random numbers.

There is also a substantial body of literature that is often grouped under the heading of stochastic search which addresses problems where we are searching for the best of a continuous set of parameters. For a thorough review of this field, see **?**. We note, however, that this field covers numerous algorithms such as stochastic approximation methods that we would not classify as falling under the heading of optimal learning.

5 Evaluating policies

When faced with an array of different learning policies, we have to address the problem of evaluating policies. Using the setting of online problems, we start by observing that we generally cannot compute the expectation in (**??**) exactly, but this equation hints at how we might evaluate a policy in practice. Let ω index both a prior $\mu(\omega)$ and a sequence of observations of $W^1(\omega), W^2(\omega), \ldots, W^N(\omega)$ which depend on the prior, where

$$W_x^n(\omega) = \mu_x(\omega) + \varepsilon_x^n(\omega).$$

Recall that $X^{\pi,n}(S^n(\omega))$ is our decision rule (policy) for choosing an alternative to test given our state of knowledge S^n. Given a sample path ω, a sample realization of a measurement policy would be computed using

$$F^\pi(\omega) = \sum_{n=0}^{N} \gamma^n \mu_{X^{\pi,n}(S^n(\omega))}(\omega). \tag{10}$$

Finally, we can repeat this simulation K times to form a statistical estimate of the value of a policy

$$\bar{F}^\pi = \frac{1}{K} \sum_{k=1}^{K} \sum_{n=0}^{N} \gamma^n \mu_{X^{\pi,n}(S^n(\omega^k))}(\omega^k). \tag{11}$$

Now we use standard statistical tools to compare policies.

We view this method of evaluating policies as fundamentally Bayesian, since we depend on a prior to determine the set of truths. However, the policy we are testing could be frequentist. A frequentist testing strategy might involve simulating a policy (Bayesian or frequentist) on a single function, where we start by assuming that we have no knowledge of the function (we can treat it initially as a constant), and where measurements of the function are noisy. While this strategy is not uncommon, it introduces dangers. For example, it may be possible to choose a policy that works well on a particular function. This problem can be mitigated by running the algorithm on a family of functions, but then this is comparable to generating a truth (the set of truths would be given by the family of functions).

6 The knowledge gradient policy

A simple idea for guiding information collection is to choose to measure the alternative which provides the greatest value from a single measurement. This idea was first introduced for ranking and selection problems by **?** as the $(R_1, \ldots, R_1)$ procedure. It has since been studied in greater depth as the knowledge gradient by **?** for problems where the measurement noise is known, and by **?** under the name of $LL(1)$ (linear loss with batch size 1) for the case where the measurement noise is unknown. The idea was recently applied to online problems (see **?**) for multiarmed bandit problems with both independent and correlated beliefs.

The "breakthrough," if it can be called this, with the idea of the knowledge gradient is the growing amount of empirical evidence that it appears to work well, even when competing against optimal policies such as Gittins indices for classical multiarmed bandit problems. This seems surprising, given that optimal learning problems can be formulated as dynamic programs. As a general statement, myopic policies often work poorly for dynamic problems, and the knowledge gradient is effectively a myopic heuristic. The insight that our numerical work appears to be showing is that while this expectation is certainly true in the context of dynamic problems that arise in the management of physical resources, it does not appear to hold true in the context of learning problems.

6.1 The knowledge gradient for off-line learning

As before, we assume that all beliefs are normally distributed with parameters captured by the state $S^n = (\mu_x^n, \beta_x^n)_{x \in \mathcal{X}}$. Given S^n, which means given our beliefs about each choice μ_x^n, the value of our current state of knowledge is given by

$$V^n(S^n) = \max_{x' \in \mathcal{X}} \mu_{x'}^n.$$

Now assume we choose to measure $x^n = x$. This means we get to observe W_x^{n+1} and update our belief about μ_x. We write this updating process using our transition equation $S^{n+1}(x) = S^M(S^n, x^n, W^{n+1})$. This means applying the Bayesian updating equations (**??**)-(**??**), but only for choice x. With this new information, the value of our new state of knowledge S^{n+1} would be given by

$$V^{n+1}(S^{n+1}(x)) = \max_{x' \in \mathcal{X}} \mu_{x'}^{n+1}.$$

At iteration n, however, the observation W^{n+1} is a random variable, which means that $V^{n+1}(S^{n+1}(x))$ is a random variable. We can compute the expected value of measuring x as

$$\nu_x^{KG,n} = \mathbb{E}\left[V^{n+1}(S^{n+1}(x)) - V^n(S^n) | S^n\right]. \tag{12}$$

We refer to $\nu_x^{KG,n}$ as the *knowledge gradient* since it is the marginal value of information from measuring x. Note that the knowledge gradient captures not only what we learn about the mean of a choice, but also the change in the precision of our belief. We write the knowledge gradient policy as choosing to measure the alternative x which has the highest marginal value of information. We

write this policy simply as

$$X^{KG,n} = \arg\max_{x \in \mathcal{X}} \nu_x^{KG,n}. \tag{13}$$

We start by computing the variance of the change in our estimate of μ_x given our state of knowledge S^n, given by

$$\tilde{\sigma}_x^{2,n} = Var[\mu_x^{n+1} - \mu_x^n | S^n].$$

It is fairly straightforward to show that

$$\begin{aligned} \tilde{\sigma}_x^{2,n} &= \bar{\sigma}_x^{2,n} - \bar{\sigma}_x^{2,n+1}, \\ &= \frac{(\bar{\sigma}_x^{2,n})}{1 + \sigma_\epsilon^2 / \bar{\sigma}_x^{2,n}}. \end{aligned} \tag{14}$$

We then compute the distance between our current estimate of the value of x, and the best of the rest, normalized by the number of standard deviations in the measurement of the change, given by

$$\zeta_x^n = -\left| \frac{\mu_x^n - \max_{x' \neq x} \mu_{x'}^n}{\tilde{\sigma}_x^n} \right|.$$

Next, we use a standard formula for $\mathbb{E} \max\{0, Z + \zeta\}$ where Z is the standard normal deviate. This formula is given by

$$f(\zeta) = \zeta \Phi(\zeta) + \phi(\zeta),$$

where $\Phi(\zeta)$ and $\phi(\zeta)$ are, respectively, the cumulative standard normal distribution and the standard normal density. Finally, the knowledge gradient is given by

$$\nu_x^{KG,n} = \tilde{\sigma}_x^n f(\zeta_x^n).$$

For a more detailed development of these equations, see **?**. Table **??** provides an illustration of the calculations for a simple problem with five alternatives.

Measurement	μ	$\bar{\sigma}$	β	$\tilde{\sigma}$	ζ	$f(\zeta)$	$\nu_x^{KG,n}$
1	20.0	18.00	0.0031	17.999	-0.444	0.215	3.878
2	22.0	12.00	0.0069	11.998	-0.500	0.198	2.373
3	24.0	25.00	0.0016	24.999	-0.160	0.324	8.101
4	26.0	12.00	0.0069	11.998	-0.167	0.321	3.853
5	28.0	16.00	0.0039	15.999	-0.125	0.340	5.432

Table 1: Calculations illustrating the knowledge gradient index