

Lecture 12

Prof. Nick Harvey

University of British Columbia

In this lecture we discuss the topic of **derandomization** — converting a randomized algorithm into a deterministic one.

1 Method of Conditional Expectations

One of the simplest methods for derandomizing an algorithm is the “method of conditional expectations”. In some contexts this is also called the “method of conditional probabilities”.

Exhaustive search. Let us start with a simple example. Let $[k]$ denote $\{1, \dots, k\}$. Suppose X is a random variable taking values in $[k]$. Let $f : [k] \rightarrow \mathbb{R}$ be any function and suppose $\mathbb{E}[f(X)] \leq \mu$. How can we find an $x \in [k]$ such that $f(x) \leq \mu$? Well, the assumption $\mathbb{E}[f(X)] \leq \mu$ guarantees that there *exists* $x \in [k]$ with $f(x) \leq \mu$. So we can simply use **exhaustive search** to try all possible values for x in only $O(k)$ time. The same idea can also be used to find an x with $f(x) \geq \mu$.

High-dimensional exhaustive search. Now let’s make the example a bit more complicated. Suppose X_1, \dots, X_n are independent random variables taking values in $[k]$. Let $f : [k]^n \rightarrow \mathbb{R}$ be any function and suppose $\mathbb{E}[f(X_1, \dots, X_n)] \leq \mu$. How can we find a vector $(x_1, \dots, x_n) \in [k]^n$ with $f(x_1, \dots, x_n) \leq \mu$? Exhaustive search is again an option, but now it will take $O(k^n)$ time, which might be too much.

Before going any further, let us define the notation

$$\mathbb{E}_{X_{i+1}, \dots, X_n} [f(x_1, \dots, x_i, X_{i+1}, \dots, X_n)] := \mathbb{E}[f(X_1, \dots, X_n) \mid X_1 = x_1, \dots, X_i = x_i]. \quad (1)$$

Usually we will consider the case where X_1, \dots, X_n are mutually independent, so the conditioning does not affect the distribution of X_{i+1}, \dots, X_n .

Method of conditional expectations. The method of conditional expectations gives a more efficient solution than high-dimensional exhaustive search, under some additional assumptions. This method dates back to a 1973 paper of [Erdos and Selfridge](#). Suppose that for any numbers (x_1, \dots, x_i) we can efficiently evaluate (1). Then the following algorithm will produce a point x_1, \dots, x_n with $f(x_1, \dots, x_n) \leq \mu$.

First we claim that the algorithm will terminate (i.e., the repeat loop will eventually succeed). To see this, define

$$g(y) = \mathbb{E}_{X_{i+1}, \dots, X_n} [f(x_1, \dots, x_{i-1}, y, X_{i+1}, \dots, X_n)].$$

Just like in our simple example above, there *exists* an x_i with

$$g(x_i) \leq \mathbb{E}_{X_i} [g(X_i)] = \mathbb{E}_{X_i, \dots, X_n} [f(x_1, \dots, x_{i-1}, X_i, X_{i+1}, \dots, X_n)].$$

so we can find such an x_i by exhaustive search. That is exactly what the repeat loop is doing.

Algorithm 1: The method of conditional expectations for finding $(x_1, \dots, x_n) \in [k]^n$ with $f(x_1, \dots, x_n) \leq \mathbb{E}[f(X_1, \dots, X_n)]$.

```
1 for  $i = 1, \dots, n$  do
2   | Set  $x_i \leftarrow 0$ 
3   | repeat
4   |   | Set  $x_i \leftarrow x_i + 1$ 
5   | until  $\mathbb{E}[f(x_1, \dots, x_i, X_{i+1}, \dots, X_n)] \leq \mathbb{E}[f(x_1, \dots, x_{i-1}, X_i, \dots, X_n)]$ ;
```

1.1 Example: Max Cut

To illustrate this method, let us consider our algorithm for the Max Cut problem from Lecture 1. We are given a graph $G = (V, E)$. Recall that this algorithm generates a cut $\delta(U)$ simply by picking a set $U \subseteq V$ uniformly at random. Equivalently, for each vertex $v \in V$, the algorithm independently flips a fair coin to decide whether to put $v \in U$. We argued that $\mathbb{E}[|\delta(U)|] \geq |E|/2$.

We will use the method of conditional expectations to derandomize this algorithm. Let the vertex set of the graph be $V = [n]$. Let

$$f(x_1, \dots, x_n) = |\delta(U)| \quad \text{where} \quad U = \{i : x_i = 1\}.$$

Let X_1, \dots, X_n be independent random variables where each X_i is 0 or 1 with probability $1/2$. We identify the event “ $X_i = 1$ ” with the event “vertex $i \in U$ ”. Then $\mathbb{E}[f(X_1, \dots, X_n)] = \mathbb{E}[|\delta(U)|] = |E|/2$. We wish to deterministically find values x_1, \dots, x_n for which $f(x_1, \dots, x_n) \geq |E|/2$.

To apply the method of conditional probabilities we must be able to efficiently compute

$$\mathbb{E}_{X_{i+1}, \dots, X_n} [f(x_1, \dots, x_i, X_{i+1}, \dots, X_n)],$$

for any numbers (x_1, \dots, x_i) . What is this quantity? It is the expected number of edges cut when we have already decided which vertices amongst $\{1, \dots, i\}$ belong to U , and the remaining vertices $\{i+1, \dots, n\}$ are placed in U randomly (independently, with probability $1/2$). This expectation is easy to compute! For any edge with both endpoints in $\{1, \dots, i\}$ we already know whether it will be cut or not. Every other edge has probability exactly $1/2$ of being cut. So we can compute that expected value in linear time.

In conclusion, the method of condition expectations gives us a deterministic, polynomial time algorithm outputting a set U with $|\delta(U)| \geq |E|/2$.

References: [Vadhan](#) Section 3.4.2.

2 Method of Pessimistic Estimators

So far we have derandomized our very simple Max Cut algorithm, which doesn't use any sophisticated probabilistic tools. Next we will see what happens when we try to apply these ideas to algorithms that use the Chernoff bound.

Let X_1, \dots, X_n be independent random variables in $[0, 1]$. Define the function f as follows:

$$f(x_1, \dots, x_n) = \begin{cases} 1 & \text{if } \sum_i x_i \geq \alpha \\ 0 & \text{if } \sum_i x_i < \alpha \end{cases}.$$

So

$$\mathbb{E}[f(X_1, \dots, X_n)] = \Pr[\sum_i X_i \geq \alpha],$$

which is the typical sort of quantity to which one would apply a Chernoff bound.

Can we apply the method of conditional expectations to this function f ? For any numbers (x_1, \dots, x_i) , we need to efficiently evaluate

$$\mathbb{E}_{X_{i+1}, \dots, X_n} [f(x_1, \dots, x_i, X_{i+1}, \dots, X_n)] = \Pr[\sum_i X_i \geq \alpha \mid X_1 = x_1, \dots, X_i = x_i].$$

Unfortunately, computing this is not so easy. If the X_i 's were i.i.d. Bernoullis then we could compute that probability by expanding it in terms of binomial coefficients. But in the non-i.i.d. or non-Bernoulli case, there does not seem to be an efficient way to compute this probability.

Here is the main idea of “pessimistic estimators”: instead of defining f to be *equal* to that probability, we will define f to be an **easily-computable upper-bound** on that probability. Because f is an upper bound on the probability of the bad event “ $\sum_i X_i \geq \alpha$ ”, the function f is called a **pessimistic estimate** of that probability. So what upper bound should we use? The Chernoff bound, of course!

For simplicity, suppose that X_1, \dots, X_n are independent Bernoulli random variables. The first step of the Chernoff bound (exponentiation and Markov's inequality) shows that, for any parameter $t > 0$,

$$\Pr[\sum_i X_i \geq \alpha] \leq \mathbb{E} \left[e^{-t\alpha} \cdot \prod_{j=1}^n e^{tX_j} \right]. \quad (2)$$

Important Remark. This step holds for *any* joint distribution on the X_i 's, including any non-independent or conditional distribution. This is because we have only used exponentiation and Markov's inequality, which need no assumptions on the distribution.

We will use the upper bound in (2) to define our function f . Specifically, define

$$f(x_1, \dots, x_n) = e^{-t\alpha} \cdot \prod_{j=1}^n e^{tx_j}. \quad (3)$$

Let's check that the conditional expectations are easy to compute with this new definition of f . Given any numbers (x_1, \dots, x_i) , we have

$$\begin{aligned} \mathbb{E}_{X_{i+1}, \dots, X_n} [f(x_1, \dots, x_i, X_{i+1}, \dots, X_n)] &= e^{-t\alpha} \cdot \prod_{j=1}^i e^{tx_j} \cdot \mathbb{E}_{X_{i+1}, \dots, X_n} \left[\prod_{j=i+1}^n e^{tX_j} \right] \\ &= e^{-t\alpha} \cdot \prod_{j=1}^i e^{tx_j} \cdot \prod_{j=i+1}^n \mathbb{E}_{X_j} [e^{tX_j}]. \end{aligned}$$

This expectation is easy to compute in linear time, assuming we know the distribution of each X_j (i.e., we know that $\Pr[X_j = 1] = p_j$).

Applying the method of conditional expectations to the pessimistic estimator. Now we'll see how to use this function f to find (x_1, \dots, x_n) with $\sum_i x_i < \alpha$. Set $\mu = \sum_i \mathbb{E}[X_i]$, $\alpha = (1 + \delta)\mu$ and $t = \ln(1 + \delta)$. We have

$$\Pr[\sum_i X_i > (1 + \delta)\mu] \leq \mathbb{E}[f(X_1, \dots, X_n)] \leq \exp\left(-\mu((1 + \delta) \ln(1 + \delta) - \delta)\right),$$

where the first inequality is from (2) and the second inequality comes from the remainder of our Chernoff bound proof. Suppose μ and δ are such that this last quantity is strictly less than 1. Then we know that there *exists* a vector (x_1, \dots, x_n) with $\sum_i x_i < \alpha$.

We now explain how to efficiently and deterministically find such a vector. The method of conditional expectations will give us a vector (x_1, \dots, x_n) for which $f(x_1, \dots, x_n) < 1$. We now apply the same argument as in (2) to a conditional distribution:

$$\begin{aligned} \Pr[\sum_i X_i \geq (1 + \delta)\mu \mid X_1 = x_1, \dots, X_n = x_n] &\leq \mathbb{E}[f(X_1, \dots, X_n) \mid X_1 = x_1, \dots, X_n = x_n] \\ &= f(x_1, \dots, x_n) < 1. \end{aligned}$$

But, under the conditional distribution “ $X_1 = x_1, \dots, X_n = x_n$ ”, there is no randomness remaining. The sum $\sum_i X_i$ is not a random variable; it is simply the number $\sum_i x_i$. Since the event “ $\sum_i x_i \geq (1 + \delta)\mu$ ” has probability less than 1, it must have probability 0. In other words, we must have $\sum_i x_i < (1 + \delta)\mu$.

This example is actually quite silly. If we want to achieve $\sum_i x_i < \alpha$, the best thing to do is obviously to set each $x_i = 0$. But the method is useful because we can apply it in more complicated scenarios that involve multiple Chernoff bounds.

2.1 Congestion Minimization

In Lecture 3 we gave a randomized algorithm which gives a $O(\log n / \log \log n)$ approximation to the congestion minimization problem. We now get a deterministic algorithm by the method of pessimistic estimators.

Recall that an instance of the problem consists of a directed graph $G = (V, A)$ with $n = |V|$ and a sequence $(s_1, t_1), \dots, (s_k, t_k)$ of pairs of vertices. We want to find s_i - t_i paths such that each arc a is contained in few paths. Let \mathcal{P}_i be the set of *all* paths in G from s_i to t_i . For every path $P \in \mathcal{P}_i$, we create a variable x_P^i .

We obtain a fractional solution to the problem by solving this LP.

$$\begin{aligned} \min \quad & C \\ \text{s.t.} \quad & \sum_{P \in \mathcal{P}_i} x_P^i = 1 \quad \forall i = 1, \dots, k \\ & \sum_i \sum_{P \in \mathcal{P}_i \text{ with } a \in P} x_P^i \leq C \quad \forall a \in A \\ & x_P^i \geq 0 \quad \forall i = 1, \dots, k \text{ and } P \in \mathcal{P}_i \end{aligned}$$

Let C^* be the optimal value of the LP.

We showed how randomized rounding gives us an integer solution (i.e., an actual set of paths). The algorithm chooses exactly one path P_i from \mathcal{P}_i by setting $P_i = P$ with probability x_P^i . For every arc a let Y_i^a be the indicator of the event “ $a \in P_i$ ”. Then the congestion on arc a is $Y^a = \sum_i Y_i^a$. We showed that $\mathbb{E}[Y^a] \leq C^*$. Let $\alpha = 6 \log n / \log \log n$. We applied Chernoff bounds to every arc and a union bound to show that

$$\Pr[\text{any } a \text{ has } Y^a > \alpha C^*] \leq \sum_{a \in A} \Pr[Y^a > \alpha C^*] \leq 1/n.$$

We will derandomize that algorithm with the function

$$f(P_1, \dots, P_k) = \sum_{a \in A} e^{-\alpha C^*} \cdot \prod_{j=1}^k e^{t_j Y_j^a}.$$

How did we obtain this function? For each arc a we applied a Chernoff bound, so each arc a has a pessimistic estimator as in (3). We add all of those functions to give us this function f .

Note that

$$\Pr[\text{any } a \text{ has } Y^a > \alpha C^*] \leq \sum_{a \in A} \Pr[Y^a > \alpha C^*] \leq \mathbb{E}[f(P_1, \dots, P_k)] \leq 1/n.$$

Applying the method of conditional expectations, we can find a vector of paths (p_1, \dots, p_k) for which $f(p_1, \dots, p_k) \leq 1/n$. Thus,

$$\begin{aligned} \Pr[\text{any } a \text{ has } Y^a > \alpha C^* \mid P_1 = p_1, \dots, P_k = p_k] &\leq \mathbb{E}[f(P_1, \dots, P_k) \mid P_1 = p_1, \dots, P_k = p_k] \\ &= f(p_1, \dots, p_k) \leq 1/n. \end{aligned}$$

Under that conditional distribution there is no randomness left, so the event “any a has $Y^a > \alpha C^*$ ” must have probability 0. So, if we choose the paths p_1, \dots, p_k then every arc a has congestion at most αC^* , as desired.