University of Washington Math 523A Lecture 6

Lecturer: Yuval Peres

Friday, April 17, 2009

1 Expected hitting times for words

1.1 Review of Li's martingale method

Setting: $X_1, X_2, \ldots$ are IID and take values in some finite alphabet $\mathcal{A}$, with

$$\mathbb{P}(X_i = a) = p_a \quad \forall a \in \mathcal{A}.$$

Given a word $w \in \mathcal{A}^k$, the hitting time of w is

$$\tau_w = \min\{t \geq 1 : (X_{t-k+1}, \ldots, X_t) = w\}.$$

(Note that we must in fact have $\tau_w \geq k$ if the sequence starts at time 1. In the next section we consider a generalized setting in which the sequence may start before time 1 and hence τ_w can be less than k.)

Last time we computed $\mathbb{E}\tau_w$ using the following martingale scheme.

Martingale method (Li 1980)

Think of a gambler, Glinda, making a sequence of fair bets on w coming up, starting at time t. First Glinda bets on the 1st digit of w, then continues to bet on each successive digit in a way that makes all the bets fair (i.e. her expected winnings are 0 at each step):

- At time t, Glinda puts a dollar on $X_t = w_1$. She gets 0 if wrong, gets $\frac{1}{p_{w_1}}$ if right. In the latter case, she bets this on $X_{t+1} = w_k$, and gets $\frac{1}{p_{w_1}p_{w_2}}$ if right.
- Continue this process $\forall j < k$: If $(X_t \ldots, X_{t+j-1}) = (w_1, \ldots, w_j)$, Glinda receives $\prod_{i=1}^{j} \frac{1}{p_{w_i}}$ and bets that on $X_{t+j} = w_j$, and she wins $\prod_{i=1}^{j+1} \frac{1}{p_{w_i}}$ if right.

Assume that one gambler enters the game at each time step, and each gambler bets on w using the scheme above. Each gambler remains in play until making a wrong bet, at which point he or she leaves the game with a net loss of one dollar. The game stops at time τ_w, at which point any gambler who is still in play wins some money if their current bet agrees

with the last digit of w. Let M_t be the net winnings (at time t) of all gamblers who entered by time t. M_t is an $\{\mathcal{F}_t\}$-martingale, where $\mathcal{F}_t = \sigma(X_1, \ldots, X_t)$. For all $t \leq \tau_w$, we have

$$M_t = \sum_{j=1}^{k} \left(\mathbf{1}_{\left\{(X_{t-j+1},\ldots,X_t)=(w_1,\ldots,w_j)\right\}} \cdot \prod_{i=1}^{j} \frac{1}{p_{w_i}} \right) - t.$$

At time t, there are t gamblers in the game, and each has paid a dollar, hence the "$-t$" term, and the sum computes their gross winnings.

Applying the Optional Stopping Theorem to M_t (after verifying that it satisfies an appropriate hypothesis) we see that

$$0 = \mathbb{E}M_0 = \mathbb{E}M_{\tau_w} = w * w - \mathbb{E}\tau_w,$$

where

$$w * w := \sum_{j=1}^{k} \left(\mathbf{1}_{\left\{(w_{k-j+1},\ldots,w_k)=(w_1,\ldots,w_j)\right\}} \cdot \prod_{i=1}^{j} \frac{1}{p_{w_i}} \right),$$

so $\mathbb{E}\tau_w = w * w$. Note that $w * w$ is the amount paid to the gamblers who win their bets when w appears.

1.2 Conditioning on an initial word

Now suppose we have two words, $v \in \mathcal{A}^\ell$ and $w \in \mathcal{A}^k$. We want to compute

$$\mathbb{E}_v\tau_w := \mathbb{E}[\tau_w \mid \text{the initial word in the sequence is } v].$$

More precisely, we place v in front of the sequence $X_1, X_2, \ldots$ so that the final digit of v occurs at time 0, and the previous digits of v correspond to negative times. Equivalently, we can define $X_{-t} = v_{\ell-t}$ for $0 \leq t \leq \ell - 1$. Depending on the final digits of v, the word w may now appear before time k (but we only count its appearance after time 0). For example, take $\mathcal{A} = \{0, 1, 2\}$, $v = 122122$, and $w = 221$. For the sequence

$$12001\underbrace{221}_{w}0012\ldots,$$

we have $\tau_w = 8$. For the same sequence with v pre-appended,

$$1221\underbrace{22|1}_{w}20012210012\ldots,$$

we have $\tau_w = 1$.

To deal with this situation we imagine that, using the martingale scheme from the previous section, the gamblers start betting at the beginning of the augmented sequence containing v. Thus, one gambler enters the game at each time step, starting at time $1 - \ell$ rather than at time 1. However, we don't count their winnings or losses from the past; we only count the bets made after time 0. If the final digits of v coincide with the initial digits of w (but do

not contain all of w), then there will still be gamblers in play at time 0 who can make bets on w starting at time 1. The bets placed by these gamblers after time 0 can be computed by looking at all ways to place w so that the beginning part of w coincides with the end part of v. We denote this amount by $v * w$:

$$v * w = \sum_{j=1}^{k} \left(\mathbf{1}_{\left\{(v_{\ell-j+1},\ldots,v_\ell)=(w_1,\ldots,w_j)\right\}} \cdot \prod_{i=1}^{j} \frac{1}{p_{w_i}} \right).$$

Note that the martingale M_t from the previous section can be written

$$M_t = (X_1, \ldots, X_t) * w - t.$$

If v does not end with the word w, define for $t \geq 0$,

$$M_t^v = (v_1, \ldots, v_\ell, X_1, \ldots, X_t) * w - (t + v * w).$$

Then M_t^v computes the net winnings of the gamblers at time $t \geq 0$ when the initial sequence is v. (The term $(v_1, \ldots, v_\ell, X_1, \ldots, X_t) * w$ computes the gross winnings at time t. The losses are 1 dollar for each of the t gamblers who entered the game since time 0, plus $v * w$ for the gamblers who were in play at time 0.) Since the game is fair, M_t^v is a martingale. It can be verified that we can apply the Optional Stopping Theorem:

$$0 = \mathbb{E}_v M_0^v = \mathbb{E}_v M_{\tau_w}^v = w * w - (\mathbb{E}_v \tau_w + v * w),$$

so

$$\mathbb{E}_v \tau_w = w * w - v * w$$

for any v that does not end with w. The case when v is the empty word $\emptyset$ reduces to the result in the previous section. Thus

$$\mathbb{E}_\emptyset \tau_w = \mathbb{E} \tau_w = w * w.$$

1.3 Examples

Let $\mathcal{A} = \{0, 1, 2\}$, and suppose each element of $\mathcal{A}$ is equally likely, so the vector of probabilities is $p(\cdot) = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$.

Recall that $w * w$ is the amount paid out when w comes up. Let $w = 221$. Since the end of w doesn't have any overlaps with the beginning of w, only one gambler wins anything when w appears: At time τ_w, the gambler who entered at time $\tau_w - 2$ has already made two successful bets, and wins $\frac{1}{\left(\frac{1}{3}\right)^3} = 27$ with the final bet on on the last digit of w. Any gambler who entered before this time must have already lost because τ_w is the first time w appears. The gamblers entering at times $\tau_w - 1$ and τ_w are both still in play at time τ_w, and they both bet on 2 at this time, but they both lose because 1 comes up. Thus $221 * 221 = 27$, and applying optional stopping at time τ_{221} gives $\mathbb{E}_\emptyset \tau_{221} = 27$.

Now suppose $w = 222$. This time w has self-overlaps, so more than one gambler wins when w appears: At time τ_w, the three gamblers who entered at times $\tau_w - 2$, $\tau_w - 1$, and

τ_w are all still in play and all bet on 2 for the next digit. When 2 appears at time τ_w, they win \$27, \$9, and \$3, respectively. Therefore, since τ_w is the total amount that has been paid by all the gamblers since time 0, applying optional stopping at time τ_w gives

$$0 = \mathbb{E}_\emptyset(\text{net gains of all gamblers at time } \tau_w) = \mathbb{E}_\emptyset(27 + 9 + 3 - \tau_{222}),$$

so $\mathbb{E}_\emptyset \tau_{222} = 222 * 222 = 27 + 9 + 3 = 39$.

Now let $v = 1122$ and $w = 221$, and consider betting on w conditioned on the initial word being v. Again we have $w * w = 27$ for the one gambler who wins when w appears, so the gross winnings at time τ_w are \$27. However, since there are two ways to place w so that the beginning of w coincides with the end of v, there are two gamblers in play at time 0, and we have to take their losses into account when computing the net winnings: At time 1, the gamblers who entered at time -1 and time 0 are still in play, and make bets of \$9 on 1, and \$3 on 2, respectively. (This shows that $v * w = 9 + 3$.) Therefore, the total losses at time τ_w are $9 + 3 = 12$ for these two gamblers, plus τ_w for all the gamblers who entered after time 0. Therefore, by optional stopping,

$$0 = \mathbb{E}_v(\text{net gains of all gamblers at time } \tau_w) = \mathbb{E}_v(27 - [\tau_w + 9 + 3]),$$

so $\mathbb{E}_{1122} \tau_{221} = 221 * 221 - 1122 * 221 = 27 - 12 = 15$.

1.4 Multiple patterns

Now suppose we are given patterns (words) $w_1, \ldots, w_n$ such that no word is a suffix of another. For example the pair $\{122, 112\}$ is allowed, but $\{122, 1122\}$ is not.

Goals:

- Find $e_{\emptyset *} := \mathbb{E}_\emptyset \min_{1 \le j \le n} \tau_{w_j}$.
- Find $\alpha_{\emptyset i} := \mathbb{P}_\emptyset \left(\tau_{w_i} = \min_{1 \le j \le n} \tau_{w_j} \right)$.

Observe that $\sum_{i=1}^n \alpha_{\emptyset i} = 1$ because, by the assumption that no word is a suffix of another, there can be no ties when one of the words comes up. Define

$$e_{ji} = \mathbb{E}_{w_j} \tau_{w_i} \ (i \neq j), \qquad e_{ii} = \mathbb{E}_{w_i} \tau_{w_i} = 0 \ (\text{the waiting time is 0}), \qquad \text{and} \quad e_{\emptyset i} = \mathbb{E}_\emptyset \tau_{w_i}.$$

From the above definitions, we have

$$\begin{aligned} e_{\emptyset i} &= e_{\emptyset *} + \sum_{j=1}^n \alpha_{\emptyset j} e_{ji} \\ &= \mathbb{E}_\emptyset \min_{1 \le k \le n} \tau_{w_k} + \sum_{j=1}^n \mathbb{P}_\emptyset \left(\tau_{w_j} = \min_{1 \le k \le n} \tau_{w_k} \right) e_{ji} \end{aligned}$$

since, starting with the empty word $\emptyset$, we have

$$\tau_{w_i} = \min_{1 \le k \le n} \tau_{w_k} + \sum_{j=1}^{n} \mathbf{1}_{\left\{\tau_{w_j} = \min_{1 \le k \le i} \tau_{w_k}\right\}} \cdot \tilde{\tau}_{w_i}$$

(where $\tilde{\tau}_{w_i} = \tau_{w_i} - \min_k \tau_{w_k}$). We can encode this in a matrix multiplication:

$$\begin{pmatrix} 0 & 1 & 1 & \dots & 1 \\ 1 & 0 & e_{21} & \dots & e_{n1} \\ 1 & e_{12} & 0 & \dots & e_{n2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e_{1n} & e_{2n} & \dots & 0 \end{pmatrix} \begin{pmatrix} e_{\emptyset *} \\ \alpha_{\emptyset 1} \\ \alpha_{\emptyset 2} \\ \vdots \\ \alpha_{\emptyset n} \end{pmatrix} = \begin{pmatrix} 1 \\ e_{\emptyset 1} \\ e_{\emptyset 2} \\ \vdots \\ e_{\emptyset n} \end{pmatrix}. \tag{1.1}$$

Here the matrix is $(n+1) \times (n+1)$, with rows and columns indexed from 0 to n. For $i, j \ge 1$, the entry of the matrix in row i, column j is e_{ji}. Recall that $e_{ji} = w_i * w_i - w_j * w_i$ and $e_{\emptyset i} = w_i * w_i$.

Fact: The matrix on the left side of (1.1) is always nonsingular. (Exercise: Find a *nice* proof of this.) Thus, the above system can be solved for $(e_{\emptyset *}, \alpha_{\emptyset 1}, \dots, \alpha_{\emptyset n})$.

2 Another application: Boolean functions

2.1 Monotone Boolean functions, e.g. recursive majority

Consider a Boolean function $f : \{1, -1\}^n \to \{1, -1\}$ mapping n bits to 1 bit. Place a partial ordering on elements $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_n)$ in $\{1, -1\}^n$ by comparing the digits coordinatewise: $x \le y$ iff $x_i \le y_i$ for $1 \le i \le n$. (Note that this is only a *partial* ordering because not every pair of elements in $\{1, -1\}^n$ is comparable.)

Definition 2.1. *A Boolean function $f : \{1, -1\}^n \to \{1, -1\}$ is* **monotone** *(increasing) if $x \le y \Rightarrow f(x) \le f(y)$.*

An example of a monotone Boolean function is the m-fold *recursive majority* of depth h, $f_m^{(h)}$, where m is odd and $h \in \mathbb{N}$. The function $f_m^{(h)} : \{1, -1\}^{m^h} \to \{1, -1\}$ is defined by recursively taking the majority of m bits. For example, take $m = 3$. The function $f_3^{(1)} : \{1, -1\}^3 \to \{1, -1\}$ computes the majority of 3 bits. The function $f_3^{(2)} : \{1, -1\}^9 \to \{1, -1\}$ is defined by breaking the input string into three blocks of 3 bits each, finding the majority bit in each of these three blocks separately, and then taking the majority of the resulting 3 bits. In general, $f_3^{(h+1)}$ is defined recursively for $h \ge 1$ by

$$f_3^{(h+1)} = f_3^{(1)} \circ \left(f_3^{(h)} \times f_3^{(h)} \times f_3^{(h)} \right),$$

where $\times$ deontes the Cartesian product of functions:

$$f_3^{(h)} \times f_3^{(h)} \times f_3^{(h)}(x, y, z) = \left(f_3^{(h)}(x), f_3^{(h)}(y), f_3^{(h)}(z) \right).$$

Note that, setting $\Omega_0 = \{1, -1\}$, we have $(x, y, z) \in \Omega_0^{3^h} \times \Omega_0^{3^h} \times \Omega_0^{3^h} = \{1, -1\}^{3^{h+1}}$ and

$$\left(f_3^{(h)}(x), f_3^{(h)}(y), f_3^{(h)}(z)\right) \in \Omega_0 \times \Omega_0 \times \Omega_0 = \{1, -1\}^3,$$

so $f_3^{(h+1)}(x, y, z) = f_3^{(1)}\left(f_3^{(h)}(x), f_3^{(h)}(y), f_3^{(h)}(z)\right) \in \Omega_0 = \{1, -1\}$.

The function $f_3^{(h)}$ can be visualized as a ternary tree of depth h, with the root at the top and the leaves at the bottom. Each of the 3^h leaves at level h corresponds to one bit of input, either $+$ or $-$. Each of the 3^{h-1} nodes at level $h-1$ is assigned a $+$ or $-$ according to the majority of its three children, and so on for each successive level up to the root, which gives the final value of the function. Note that the recursive majority function does not necessarily compute the majority of the input bits. For example, consider $f_3^{(2)}$ applied to the string

$$++-++---.$$

2.2 Randomized algorithms

Consider the probability space $\Omega = \Omega_0^n = \{1, -1\}^n$ equipped with the probability measure $\mathbb{P} = \left(\frac{1}{2}, \frac{1}{2}\right)^n$. Suppose f is a monotone Boolean function on Ω. If $x = (x_1, \ldots, x_n)$ is a random element of Ω, the **influence** of variable j on f is

$$\begin{aligned} I_j(f) = \mathbb{E}[x_j f(x)] &= \mathbb{P}[f(x) = 1 \mid x_j = 1] - \mathbb{P}[f(x) = 1 \mid x_j = -1] \\ &\quad + \mathbb{P}[f(x) = -1 \mid x_j = -1] - \mathbb{P}[f(x) = -1 \mid x_j = 1]. \end{aligned}$$

(The formula follows from the definitions of expectation and conditional probability.)

We are interested in randomized algorithms to compute f exactly. An algorithm is specified by a random sequence of indices $k(1), k(2), \ldots, k(\tau)$ telling us which input variables to look at; τ is a stopping time indicating when the algorithm is finished computing. More explicitly, the index $k(j)$ is computed as some function of the previously revealed variables and an independent source of randomness:

$$k(j) = F_j\left(x_{k(1)}, x_{k(2)}, \ldots, x_{k(j-1)}, U_j\right),$$

where U_j is independent of $\{x, U_1, \ldots, U_{j-1}\}$. The time τ when the algorithm terminates is defined by

$$\tau = \min\left\{j : f(x) \text{ is determined by } x_{k(1)}, x_{k(2)}, \ldots, x_{k(j)}\right\}.$$

Thus τ is a stopping time with respect to the filtration $\mathcal{F} = (\mathcal{F}_j)$, where

$$\mathcal{F}_j = \sigma\left\{U_1, \ldots, U_j\,; x_{k(1)}, x_{k(2)}, \ldots, x_{k(j)}\right\}$$

is the information available at the jth step.

For example, let $f : \Omega_0^3 \to \Omega_0$ be the majority of three variables. Let $k(1) \in \text{Unif}\{1, 2, 3\}$ and $k(2) \in \text{Unif}(\{1, 2, 3\} \setminus \{k(1)\})$. If $x_{k(1)} = x_{k(2)}$, then $\tau = 2$ and $f(x) = x_{k(1)}$. If $x_{k(1)} \neq x_{k(2)}$, then we need to look at the 3rd bit to determine $f(x)$, and $\tau = 3$. Thus we have

$$\mathbb{P}(\tau = 2) = \frac{1}{2} \quad \text{and} \quad \mathbb{E}\tau = \frac{5}{2}.$$

Now suppose $n = 3^h$ and let $f = f_3^{(h)}$ be the 3-fold recursive majority on $\Omega = \Omega_0^n$. Then it's easy to find an algorithm with $\mathbb{E}\tau = (2.5)^h$ (e.g. think of f as a ternary tree, and at each level, choose two children at random). By comparison, the best known algorithm has $\mathbb{E}\tau = (2.46\ldots)^h$. Here's a general inequality:

> $\left(\sum_{j=1}^n I_j(f)\right)^2 \leq \mathbb{E}\tau$ for any randomized algorithm that computes a monotone $f : \{1, -1\}^n \to \{1, -1\}$ exactly.

For f = recursive majority, the influence of a variable at depth k is 2^{-k} (again, we are thinking of f as a tree, and we can think of the inner nodes as intermediate variables in the computation of f). Therefore, $I_j(f) = \left(\frac{1}{2}\right)^h$ for any j. Since $n = 3^h$ we have $\sum_{j=1}^n I_j(f) = \left(\frac{3}{2}\right)^h$, so the above inequality gives $\left(\frac{3}{2}\right)^{2h} = (2.25)^h \leq \mathbb{E}\tau$. Thus the expected running time $(2.46\ldots)^h$ for the best known algorithm is already fairly close to the best we could possibly hope for.