Chapter 2

Probability and Expectation

By Sariel Har-Peled, March 19, 2024[①]

Everybody knows that the dice are loaded
Everybody rolls with their fingers crossed
Everybody knows the war is over
Everybody knows the good guys lost
Everybody knows the fight was fixed
The poor stay poor, the rich get rich
That's how it goes
Everybody knows

Everybody knows, Leonard Cohen

2.1. Basic probability

Here we recall some definitions about probability. The reader already familiar with these definition can happily skip this section.

2.1.1. Formal basic definitions: Sample space, σ-algebra, and probability

A ***sample space*** Ω is a set of all possible outcomes of an experiment. We also have a set of events $\mathcal{F}$, where every member of $\mathcal{F}$ is a subset of Ω. Formally, we require that $\mathcal{F}$ is a σ-algebra.

Definition 2.1.1. A single element of Ω is an ***elementary event*** or an ***atomic event***.

Definition 2.1.2. A set $\mathcal{F}$ of subsets of Ω is a ***σ-algebra*** if:
(i) $\mathcal{F}$ is not empty,
(ii) if $X \in \mathcal{F}$ then $\overline{X} = (\Omega \setminus X) \in \mathcal{F}$, and
(iii) if $X, Y \in \mathcal{F}$ then $X \cup Y \in \mathcal{F}$.
More generally, we require that if $X_i \in \mathcal{F}$, for $i \in \mathbb{Z}$, then $\cup_i X_i \in \mathcal{F}$. A member of $\mathcal{F}$ is an ***event***.

As a concrete example, if we are rolling a dice, then $\Omega = \{1, 2, 3, 4, 5, 6\}$ and $\mathcal{F}$ would be the power set of all possible subsets of Ω.

Definition 2.1.3. A ***probability measure*** is a mapping $\mathbb{P} : \mathcal{F} \to [0, 1]$ assigning ***probabilities*** to events. The function $\mathbb{P}$ needs to have the following properties:
(i) Additive: for $X, Y \in \mathcal{F}$ disjoint sets, we have that $\mathbb{P}[X \cup Y] = \mathbb{P}[X] + \mathbb{P}[Y]$, and
(ii) $\mathbb{P}[\Omega] = 1$.

Observation 2.1.4. *Let $C_1, \ldots, C_n$ be random events (not necessarily independent). Than*

$$\mathbb{P}[\cup_{i=1}^n C_i] \leq \sum_{i=1}^n \mathbb{P}[C_i].$$

(This is usually referred to as the **union bound***.) If $C_1, \ldots, C_n$ are* disjoint *events then*

$$\mathbb{P}[\cup_{i=1}^n C_i] = \sum_{i=1}^n \mathbb{P}[C_i].$$

Definition 2.1.5. A ***probability space*** is a triple $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω is a sample space, $\mathcal{F}$ is a σ-algebra defined over Ω, and $\mathbb{P}$ is a probability measure.

Definition 2.1.6. A ***random variable*** f is a mapping from Ω into some set $\mathcal{G}$. We require that the probability of the random variable to take on any value in a given subset of values is well defined. Formally, for any subset $U \subseteq \mathcal{G}$, we have that $f^{-1}(U) \in \mathcal{F}$. That is, $\mathbb{P}[f \in U] = \mathbb{P}\left[f^{-1}(U)\right]$ is defined.

Going back to the dice example, the number on the top of the dice when we roll it is a random variable. Similarly, let X be one if the number rolled is larger than 3, and zero otherwise. Clearly X is a random variable.

We denote the ***probability*** of a random variable X to get the value x, by $\mathbb{P}[X = x]$ (or sometime $\mathbb{P}[x]$, if we are lazy).

2.1.2. Expectation and conditional probability

Definition 2.1.7 (Expectation). The expectation of a random variable X, is its average. Formally, the ***expectation*** of X is

$$\mathbb{E}[X] = \sum_x x\,\mathbb{P}[X = x].$$

Definition 2.1.8 (Conditional Probability.). The ***conditional probability*** of X given Y, is the probability that $X = x$ given that $Y = y$. We denote this quantity by $\mathbb{P}[X = x \mid Y = y]$.

One useful way to think about the conditional probability $\mathbb{P}[X \mid Y]$ is as a function, between the given value of Y (i.e., y), and the probability of X (to be equal to x) in this case. Since in many cases x and y are omitted in the notation, it is somewhat confusing.

The conditional probability can be computed using the formula

$$\mathbb{P}[X = x \mid Y = y] = \frac{\mathbb{P}[(X = x) \cap (Y = y)]}{\mathbb{P}[Y = y]}.$$

For example, let us roll a dice and let X be the number we got. Let Y be the random variable that is true if the number we get is even. Then, we have that

$$\mathbb{P}\left[X = 2 \,\middle|\, Y = \mathit{true}\right] = \frac{1}{3}.$$

Definition 2.1.9. Two random variables X and Y are ***independent*** if $\mathbb{P}\left[X = x \,\middle|\, Y = y\right] = \mathbb{P}[X = x]$, for all x and y.

Observation 2.1.10. *If X and Y are independent then $\mathbb{P}\left[X = x \,\middle|\, Y = y\right] = \mathbb{P}[X = x]$ which is equivalent to $\dfrac{\mathbb{P}[X = x \cap Y = y]}{\mathbb{P}[Y = y]} = \mathbb{P}[X = x]$. That is, X and Y are* independent*, if for all x and y, we have that*

$$\mathbb{P}[X = x \cap Y = y] = \mathbb{P}[X = x]\,\mathbb{P}[Y = y].$$

Remark. Informally, and not quite correctly, one possible way to think about the conditional probability $\mathbb{P}[X = x \mid Y = y]$ is that it measure the benefit of having more information. If we know that $Y = y$, do we have any change in the probability of $X = x$?

Lemma 2.1.11 (Linearity of expectation). **Linearity of expectation** *is the property that for any two random variables X and Y, we have that* $\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y]$.

Proof: $\mathbb{E}[X+Y] = \sum_{\omega\in\Omega} \mathbb{P}[\omega](X(\omega)+Y(\omega)) = \sum_{\omega\in\Omega} \mathbb{P}[\omega]X(\omega) + \sum_{\omega\in\Omega} \mathbb{P}[\omega]Y(\omega) = \mathbb{E}[X] + \mathbb{E}[Y]$. ■

Lemma 2.1.12. *If X and Y are two random independent variables, then* $\mathbb{E}[XY] = \mathbb{E}[X]\,\mathbb{E}[Y]$.

Proof: Let $U(X)$ the sets of all the values that X might have. We have that

$$
\begin{aligned}
\mathbb{E}[XY] &= \sum_{x\in U(X), y\in U(Y)} xy\,\mathbb{P}[X = x \text{ and } Y = y] = \sum_{x\in U(X), y\in U(Y)} xy\,\mathbb{P}[X = x]\,\mathbb{P}[Y = y] \\
&= \sum_{x\in U(X)} \sum_{y\in U(Y)} xy\,\mathbb{P}[X = x]\,\mathbb{P}[Y = y] = \sum_{x\in U(X)} x\,\mathbb{P}[X = x] \sum_{y\in U(Y)} y\,\mathbb{P}[Y = y] \\
&= \mathbb{E}[X]\,\mathbb{E}[Y].
\end{aligned}
$$

■

2.1.3. Variance and standard deviation

Definition 2.1.13 (Variance and Standard Deviation). For a random variable X, let

$$
\mathbb{V}[X] = \mathbb{E}\Big[(X-\mu_X)^2\Big] = \mathbb{E}\Big[X^2\Big] - \mu_X^2
$$

denote the ***variance*** of X, where $\mu_X = \mathbb{E}[X]$. Intuitively, this tells us how concentrated is the distribution of X. The ***standard deviation*** of X, denoted by σ_X is the quantity $\sqrt{\mathbb{V}[X]}$.

Observation 2.1.14. *(i) For any constant $c \geq 0$, we have* $\mathbb{V}[cX] = c^2\,\mathbb{V}[X]$.
(ii) For X and Y independent variables, we have $\mathbb{V}[X+Y] = \mathbb{V}[X] + \mathbb{V}[Y]$.

2.2. Some distributions and their moments

2.2.1. Bernoulli distribution

Definition 2.2.1 (Bernoulli distribution). Assume, that one flips a coin and get 1 (heads) with probability p, and 0 (i.e., tail) with probability $q = 1-p$. Let X be this random variable. The variable X is has ***Bernoulli distribution*** with parameter p.

We have that $\mathbb{E}[X] = 1\cdot p + 0\cdot(1-p) = p$, and

$$
\mathbb{V}[X] = \mathbb{E}\Big[X^2\Big] - \mu_X^2 = \mathbb{E}\Big[X^2\Big] - p^2 = p - p^2 = p(1-p) = pq.
$$

Definition 2.2.2 (Binomial distribution). Assume that we repeat a Bernoulli experiment n times (independently!). Let $X_1, \ldots, X_n$ be the resulting random variables, and let $X = X_1 + \cdots + X_n$. The variable X has the ***binomial distribution*** with parameters n and p. We denote this fact by $X \sim \mathrm{Bin}(n, p)$. We have

$$
b(k; n, p) = \mathbb{P}[X = k] = \binom{n}{k} p^k q^{n-k}.
$$

Also, $\mathbb{E}[X] = np$, and $\mathbb{V}[X] = \mathbb{V}\big[\sum_{i=1}^n X_i\big] = \sum_{i=1}^n \mathbb{V}[X_i] = npq$.

2.2.2. Geometric distribution

Definition 2.2.3. Consider a sequence $X_1, X_2, \ldots$ of independent Bernoulli trials with probability p for success. Let X be the number of trials one has to perform till encountering the first success. The distribution of X is a ***geometric distribution*** with parameter p. We denote this by $X \sim \mathrm{Geom}(p)$.

Lemma 2.2.4. *For a variable $X \sim \mathrm{Geom}(p)$, we have, for all i, that $\mathbb{P}[X = i] = (1-p)^{i-1}p$. Furthermore, $\mathbb{E}[X] = 1/p$ and $\mathbb{V}[X] = (1-p)/p^2$.*

Proof: The proof of the expectation and variance is included for the sake of completeness, and the reader is of course encouraged to skip (reading) this proof. So, let $f(x) = \sum_{i=0}^{\infty} x^i = \frac{1}{1-x}$, and observe that $f'(x) = \sum_{i=1}^{\infty} i x^{i-1} = (1-x)^{-2}$. As such, we have

$$\mathbb{E}[X] = \sum_{i=1}^{\infty} i\,(1-p)^{i-1}p = p f'(1-p) = \frac{p}{(1-(1-p))^2} = \frac{1}{p},$$

$$\text{and } \mathbb{V}[X] = \mathbb{E}\Big[X^2\Big] - \frac{1}{p^2} = \sum_{i=1}^{\infty} i^2\,(1-p)^{i-1}p - \frac{1}{p^2}. = p + p(1-p)\sum_{i=2}^{\infty} i^2\,(1-p)^{i-2} - \frac{1}{p^2}.$$

Observe that

$$f''(x) = \sum_{i=2}^{\infty} i(i-1)x^{i-2} = \Big((1-x)^{-1}\Big)'' = \frac{2}{(1-x)^3}.$$

As such, we have that

$$\begin{aligned}\Delta(x) &= \sum_{i=2}^{\infty} i^2 x^{i-2} = \sum_{i=2}^{\infty} i(i-1)x^{i-2} + \sum_{i=2}^{\infty} i x^{i-2} = f''(x) + \frac{1}{x}\sum_{i=2}^{\infty} i x^{i-1} = f''(x) + \frac{1}{x}(f'(x) - 1)\\ &= \frac{2}{(1-x)^3} + \frac{1}{x}\left(\frac{1}{(1-x)^2} - 1\right) = \frac{2}{(1-x)^3} + \frac{1}{x}\left(\frac{1-(1-x)^2}{(1-x)^2}\right) = \frac{2}{(1-x)^3} + \frac{1}{x}\cdot\frac{x(2-x)}{(1-x)^2}\\ &= \frac{2}{(1-x)^3} + \frac{2-x}{(1-x)^2}.\end{aligned}$$

As such, we have that

$$\begin{aligned}\mathbb{V}[X] &= p + p(1-p)\Delta(1-p) - \frac{1}{p^2} = p + p(1-p)\left(\frac{2}{p^3} + \frac{1+p}{p^2}\right) - \frac{1}{p^2} = p + \frac{2(1-p)}{p^2} + \frac{1-p^2}{p} - \frac{1}{p^2}\\ &= \frac{p^3 + 2(1-p) + p - p^3 - 1}{p^2} = \frac{1-p}{p^2}.\end{aligned}$$

■

2.3. Application of expectation: Approximating 3SAT

Let F be a boolean formula with n variables in CNF form, with m clauses, where each clause has exactly k literals. We claim that a random assignment for F, where value 0 or 1 is picked with probability $1/2$, satisfies in expectation $(1-2^{-k})m$ of the clauses.

We remind the reader that an instance of 3SAT is a boolean formula, for example $F = (x_1 + x_2 + x_3)(x_4 + \overline{x_1} + x_2)$, and the decision problem is to decide if the formula has a satisfiable assignment. Interestingly, we can turn this into an optimization problem.

Max 3SAT

Instance: A collection of clauses: $C_1, \ldots, C_m$.
Question: Find the assignment to $x_1, ..., x_n$ that satisfies the maximum number of clauses.

Clearly, since 3SAT is NP-Complete it implies that Max 3SAT is NP-Hard. In particular, the formula F becomes the following set of two clauses:

$$x_1 + x_2 + x_3 \quad \text{and} \quad x_4 + \overline{x_1} + x_2.$$

Note, that Max 3SAT is a ***maximization problem***.

Definition 2.3.1. Algorithm **Alg** for a maximization problem achieves an approximation factor α if for all inputs, we have:

$$\frac{\mathbf{Alg}(G)}{\mathrm{Opt}(G)} \geq \alpha.$$

In the following, we present a ***randomized algorithm*** – it is allowed to consult with a source of random numbers in making decisions. A key property we need about random variables, is the linearity of expectation property defined above.

Definition 2.3.2. For an event $\mathcal{E}$, let X be a random variable which is 1 if $\mathcal{E}$ occurred, and 0 otherwise. The random variable X is an ***indicator variable***.

Observation 2.3.3. *For an indicator variable X of an event $\mathcal{E}$, we have*

$$\mathbb{E}[X] = 0 \cdot \mathbb{P}[X = 0] + 1 \cdot \mathbb{P}[X = 1] = \mathbb{P}[X = 1] = \mathbb{P}[\mathcal{E}].$$

Theorem 2.3.4. *One can achieve (in expectation) $(7/8)$-approximation to Max 3SAT in polynomial time. Specifically, consider a 3SAT formula F with n variables and m clauses, and consider the randomized algorithm that assigns each variable value 0 or 1 with equal probability (independently to each variable). Then this assignment satisfies $(7/8)m$ clauses in* expectation.

Proof: Let $x_1, \ldots, x_n$ be the n variables used in the given instance. The algorithm works by randomly assigning values to $x_1, \ldots, x_n$, independently, and equal probability, to 0 or 1, for each one of the variables.

Let Y_i be the indicator variables which is 1 if (and only if) the ith clause is satisfied by the generated random assignment, and 0 otherwise, for $i = 1, \ldots, m$. Formally, we have

$$Y_i = \begin{cases} 1 & C_i \text{ is satisfied by the generated assignment,} \\ 0 & \text{otherwise.} \end{cases}$$

Now, the number of clauses satisfied by the given assignment is $Y = \sum_{i=1}^{m} Y_i$. We claim that $\mathbb{E}[Y] = (7/8)m$, where m is the number of clauses in the input. Indeed, we have

$$\mathbb{E}[Y] = \mathbb{E}\Big[\sum_{i=1}^{m} Y_i\Big] = \sum_{i=1}^{m} \mathbb{E}[Y_i],$$

by linearity of expectation. The probability that $Y_i = 0$ is exactly the probability that all three literals appearing in the clause C_i are evaluated to FALSE. Since the three literals, Say ℓ_1, ℓ_2, ℓ_3, are instance of three distinct variable these three events are independent, and as such the probability for this happening is

$$\mathbb{P}[Y_i = 0] = \mathbb{P}[(\ell_1 = 0) \wedge (\ell_2 = 0) \wedge (\ell_3 = 0)] = \mathbb{P}[\ell_1 = 0]\,\mathbb{P}[\ell_2 = 0]\,\mathbb{P}[\ell_3 = 0] = \frac{1}{2} * \frac{1}{2} * \frac{1}{2} = \frac{1}{8}.$$

(Another way to see this, is to observe that since C_i has exactly three literals, there is only one possible assignment to the three variables appearing in it, such that the clause evaluates to FALSE. Now, there are eight (8) possible assignments to this clause, and thus the probability of picking a FALSE assignment is $1/8$.) Thus,

$$\mathbb{P}[Y_i = 1] = 1 - \mathbb{P}[Y_i = 0] = \frac{7}{8},$$

and

$$\mathbb{E}[Y_i] = \mathbb{P}[Y_i = 0] * 0 + \mathbb{P}[Y_i = 1] * 1 = \frac{7}{8}.$$

Namely, $\mathbb{E}[\text{\# of clauses sat}] = \mathbb{E}[Y] = \sum_{i=1}^{m} \mathbb{E}[Y_i] = (7/8)m$. Since the optimal solution satisfies at most m clauses, the claim follows. ■

Curiously, Theorem 2.3.4 is stronger than what one usually would be able to get for an approximation algorithm. Here, the approximation quality is independent of how well the optimal solution does (the optimal can satisfy at most m clauses, as such we get a $(7/8)$-approximation. Curiouser and curiouser[②], the algorithm does not even look on the input when generating the random assignment.

Håstad [Hås01] proved that one can do no better; that is, for any constant $\varepsilon > 0$, one can not approximate 3SAT in polynomial time (unless P = NP) to within a factor of $7/8 + \varepsilon$. It is pretty amazing that a trivial algorithm like the above is essentially optimal.

Remark 2.3.5. For $k \geq 3$, the above implies $(1 - 2^{-k})$-approximation algorithm, for k-SAT, as long as the instances are each of length at least k.

2.4. Markov's inequality

2.4.1. Markov's inequality

We remind the reader that for a random variable X assuming real values, its expectation is $\mathbb{E}[Y] = \sum_y y \cdot \mathbb{P}[Y = y]$. Similarly, for a function $f(\cdot)$, we have $\mathbb{E}[f(Y)] = \sum_y f(y) \cdot \mathbb{P}[Y = y]$.

Theorem 2.4.1 (Markov's Inequality). *Let Y be a random variable assuming only non-negative values. Then for all $t > 0$, we have*

$$\mathbb{P}[Y \geq t] \leq \frac{\mathbb{E}[Y]}{t}.$$

Proof: Indeed,

$$\begin{aligned}\mathbb{E}[Y] &= \sum_{y \geq t} y\, \mathbb{P}[Y = y] + \sum_{y < t} y\, \mathbb{P}[Y = y] \geq \sum_{y \geq t} y\, \mathbb{P}[Y = y] \\ &\geq \sum_{y \geq t} t\, \mathbb{P}[Y = y] = t\, \mathbb{P}[Y \geq t].\end{aligned}$$

■

Markov inequality is tight, as the following exercise testifies.

Exercise 2.4.2. For any (integer) $k > 1$, define a random positive variable X_k such that $\mathbb{P}[X_k \geq k\, \mathbb{E}[X_k]] = 1/k$.

[②]"Curiouser and curiouser!" Cried Alice (she was so much surprised, that for the moment she quite forgot how to speak good English). – Alice in wonderland, Lewis Carol

2.4.2. Example: A good approximation to kSAT with good probability

In Section 2.3 we saw a surprisingly simple algorithm that, for a formula F that is 3SAT with n variables and m clauses, in expectation (in linear time) it finds an assignment that satisfies $(7/8)m$ of the clauses (for simplicity, here we set $k = 3$).

The problem is that the guarantee is only in expectation – and the assignment being output by the algorithm might satisfy much fewer clauses. Namely, we would like to convert a guarantee that is in expectation into, a good probability guarantee. So, let $\varepsilon, \varphi \in (0, 1/2)$ be two parameters. We would like an algorithm that outputs an assignment that satisfies (say) $(1-\varepsilon)(7/8)m$ clauses, with probability $\geq 1-\varphi$.

To this end, the new algorithm runs the previous algorithm

$$u = \left\lceil \frac{1}{\varepsilon} \ln \frac{1}{\varphi} \right\rceil$$

times, and returns the assignment satisfying the largest number of clauses.

Lemma 2.4.3. *Given a 3SAT formula with n variables and m clauses, and parameters $\varepsilon, \varphi \in (0, 1/2)$, the above algorithm returns an assignment that satisfies $\geq (1-\varepsilon)(7/8)m$ clauses of F, with probability $\geq 1-\varphi$. The running time of the algorithm is $O(\varepsilon^{-1}(n+m)\log \varphi^{-1})$.*

Proof: Let Z_i be the number of clauses *not* satisfied by the ith random assignment considered by the algorithm. Observe that $\mathbb{E}[Z_i] = m/8$,as the probability of a clause not to be satisfied is $1/2^3$. The ith iteration *fails* if

$$m - Z_i < (1-\varepsilon)(7/8)m \qquad \Longrightarrow \qquad Z_i > m(1-(1-\varepsilon)7/8) = (1+7\varepsilon)\frac{m}{8} = (1+7\varepsilon)\,\mathbb{E}[Z_i]\,.$$

Thus, by Markov's inequality, the ith iteration fails with probability

$$p = \mathbb{P}[m - Z_i < (1-\varepsilon)(7/8)m] = \mathbb{P}[Z_i > (1+7\varepsilon)\,\mathbb{E}[Z_i]] < \frac{\mathbb{E}[Z_i]}{(1+7\varepsilon)\,\mathbb{E}[Z_i]} = \frac{1}{1+7\varepsilon} < 1-\varepsilon,$$

since $(1+7\varepsilon)(1-\varepsilon) = 1 + 6\varepsilon - 7\varepsilon^2 > 1$, for $\varepsilon < 1/2$.

For the algorithm to fail, all u iterations must fail. Since $1 - x \leq \exp(-x)$, we have that

$$p^u \leq (1-\varepsilon)^u \leq \exp(-\varepsilon)^u \leq \exp(-\varepsilon u) \leq \exp\left(-\varepsilon \left\lceil \frac{1}{\varepsilon} \ln \frac{1}{\varphi} \right\rceil\right) \leq \varphi. \qquad \blacksquare$$

2.4.3. Example: Coloring a graph

Consider a graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ with n vertices and m edges. We would like to color it with k colors. As a reminder, a ***coloring*** of a graph by k colors, is an assignment $\chi : \mathsf{V} \to [\![k]\!]$ of a color to each vertex of G, out of the k possible colors $[\![k]\!] = \{1, 2, \ldots, k\}$. A coloring of an edge $uv \in \mathsf{E}$ is *valid* if $\chi(u) \neq \chi(v)$.

Lemma 2.4.4. *Consider a random coloring χ of the vertices of a graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$, where each vertex is assigned a color randomly and uniformly from $[\![k]\!]$. Then, the expected number of edges with invalid coloring is m/k, where $m = |\mathsf{E}(\mathsf{G})|$ is the number of edges in G.*

Proof: Let $\mathsf{E} = \{e_1, \ldots, e_m\}$. Let X_i be an indicator variable that is $1 \iff e_i$ is invalid colored by χ. Let $e_i = u_i v_i$. We have that

$$\mathbb{P}[X_i = 1] = \mathbb{P}[\chi(u_i) = \chi(v_i)] = \frac{1}{k}.$$

Indeed, conceptually color u_i first, and v_i later. The probability that v_i would be assigned the same color as u_i is $1/k$. Let Z be the random variable that is the number of edges that are invalid for χ. We have that $Z = \sum_i X_i$. By linearity of expectations, and the expectation of an indicator variable, we have

$$\mathbb{E}[Z] = \sum_{i=1}^{m} \mathbb{E}[X_i] = \sum_{i=1}^{m} \mathbb{P}[X_i = 1] = \sum_{i=1}^{m} \frac{1}{k} = \frac{m}{k}. \qquad \blacksquare$$

That is pretty good, but what about an algorithm that always succeeds? The above algorithm might always somehow gives us a bad coloring. Well, not to worry.

Lemma 2.4.5. *The above random coloring of* G *with* k *colors, has at most* $2m/k$ *invalid edges, with probability* $\geq 1/2$.

Proof: We have that $\mathbb{E}[Z] = m/k$. As such, by Markov's inequality, we have that

$$\mathbb{P}[Z > 2m/k] \leq \mathbb{P}[Z \geq 2m/k] \leq \frac{\mathbb{E}[Z]}{2m/k} = \frac{m/k}{2m/k} = \frac{1}{2}.$$

Thus

$$\mathbb{P}[Z \leq 2m/k] = 1 - \mathbb{P}[Z > 2m/k] \geq 1 - \frac{1}{2} = \frac{1}{2}. \qquad \blacksquare$$

In particular, consider he modified algorithm – it randomly colors the graph G. If there are at most $2m/k$ invalid edges, it output the coloring, and stops. Otherwise, it retries. The probability of every iteration to succeeds is $p \geq 1/2$, and as such, the number of iterations behaves like a geometric random variable. It follows, that in expectation, the number of iterations is at most $1/p \leq 2$. Thus, the expected running time of this algorithm is $O(m)$. Indeed, let R be the number iterations performed by the algorithm. We have that the expected running time is proportional to

$$\mathbb{E}[Rm] = m\,\mathbb{E}[R] = 2m.$$

Note, that this is not the full picture – $\mathbb{P}[R = i] \leq 1/2^{i-1}$. So the probability of this algorithm tor for long decreases quickly.

2.4.3.1. Getting a valid coloring

2.4.3.1.1. A fun algorithm. A natural approach is to run the above algorithm for $k = \sqrt{m}$ (assume it is an integer). Then identify all the invalid edges, and invalidate the color of all the vertices involved. We now repeat the coloring algorithm on these invalid vertices and invalid edges, again using random coloring, but now using colors $\{k+1, \ldots, 2k\}$. If after this, there is a single invalid edge, we color one of its vertices by the color $2k+1$, and output this coloring. Otherwise, it fails.

Lemma 2.4.6. *The above algorithm succeeds with probability at least* $1/2$.

Proof: Let Y be the number of invalid edges in the end of the second round. For an edge to be invalid, its coloring must have failed in both rounds, and the probability for that is exactly $(1/k) \cdot (1/k) = 1/m$ since the two events are independent. As such, arguing as above, we have $\mathbb{E}[Y] = 1$. By Markov's inequality, we have that

$$\mathbb{P}[\text{algorithm fails}] = \mathbb{P}[Y > 1] = \mathbb{P}[Y \geq 2] \leq \frac{\mathbb{E}[Y]}{2} = \frac{1}{2}. \qquad \blacksquare$$

Remark 2.4.7. This is a toy example - it is not hard to come up with a deterministic algorithm that uses (say) $\sqrt{2m} + 2$ colors (how? think about it). However, this algorithm is a nice distributed algorithm - after three rounds of communications, it colors the graph in a valid way, with probability at least half.

References

[Hås01] J. Håstad. Some optimal inapproximability results. *J. Assoc. Comput. Mach.*, 48(4): 798–859, 2001.