Recitation 9 — Probability and Maximal Independent Sets

Parallel and Sequential Data Structures and Algorithms, 15-210 (Fall 2013)

October 23, 2013

1 Announcements

- Assignment 7 will be out shortly.
- Today's recitation will review some probability basics and discuss maximal independent sets, as an example of some of the techniques we've been studying.

2 Probability Basics

Many of you have seen probability before. We'll quickly go through the basics and move on to more interesting things.

2.1 Conditional Probability

$P(A|B) = P(A \cap B)/P(B)$. This identity can be intuitively demonstrated on a Venn diagram.

2.2 Independence

Say I throw a fair six-sided dice three times. The probability of rolling an even number, then a four, then an even number again is $\frac{1}{24}$. This is because rolling an even number on a six-sided dice occurs with $\frac{1}{2}$ probability, and rolling a four occurs with $\frac{1}{6}$ probability, and $(\frac{1}{2})^2 \cdot \frac{1}{6} = \frac{1}{24}$. Now, say I draw three cards from a shuffled 52-card deck. What is the probability of drawing any card in the hearts suit, then the queen of hearts, then any hearts card again? This question is harder because the events here are not independent.

Formally, event A is independent from event B iff $P(A|B) = P(A)$; i.e. the probability of A remains the same whether or not B has occurred. From the identity $P(A|B) = P(A \cap B)/P(B)$, it follows that

$$P(A|B) = P(A) \iff P(A \cap B)/P(B) = P(A) \iff P(A \cap B)/P(A) = P(B) \iff P(B|A) = P(B)$$

so we see that event A being independent from event B also implies that B is independent from A.

Using the identity $P(A|B) = P(A \cap B)/P(B)$, the probability that events A, B, and C occur, i.e. $P(A \cap B \cap C)$, can be expressed as $P(A) \cdot P(B|A) \cdot P(C|A \cap B)$. If A, B, and C are mutually independent, then this is just $P(A) \cdot P(B) \cdot P(C)$, which is what made the first dice example so easy. (Note that mutual independence is a stronger condition than pairwise independence.) The second example with cards is hard because we're stuck with trying to figure out the probability of drawing the queen of hearts given that we've drawn some hearts card already, and then the probability of drawing another hearts given that we've drawn some hearts card and then the queen of hearts.

Keep in mind that events can be dependent even when they occur at the same time with the same probability. For example, consider a shell game, where a pea is hidden underneath one of three shells (with uniform probability). While each shell has a $\frac{1}{3}$ probability of containing the pea, the event that shell A contains the pea

is not independent from the event that shell B contains the pea. At most one can contain the pea; there isn't a $\frac{1}{27}$ probability that all three shells contain the pea! In this example, the events of A and B containing the pea are mutually exclusive—$P(\text{A contains the pea}|\text{B contains the pea}) = 0$, or equivalently $P(A \cap B) = 0$.

2.3 Inclusion-Exclusion

$P(A \cup B) = P(A) + P(B) - P(A \cap B)$. This principle can be intuitively demonstrated on a Venn diagram. Note that $P(A \cap B) = 0$ if and only if A and B are mutually exclusive, so be careful when directly adding probabilities.

2.4 Expected value

A random variable (usually a 1-dimensional number in the reals) is a variable representing the outcome of a random process. While ordinary variables are considered to have a single unknown value, random variables are considered to have all possible values, each with a particular probability. This isn't quite true, but it's good enough for 210, where we only deal with discrete probability.

The expected value of a random variable is the weighted mean of its possible values, where each value x is weighted by the probability that x is the outcome. For example, a random variable representing the outcomes of a fair six-sided dice has an expected value 3.5, which is the average of 1, 2, 3, 4, 5, and 6 when each is given equal weight. What would the expected value be for a dice that is unfair, such that even numbers are rolled with probability $\frac{2}{9}$ and odd numbers are rolled with probability $\frac{1}{9}$? You could do it the long way, "$1 \cdot \frac{1}{9} + 2 \cdot \frac{2}{9} + 3 \cdot \frac{1}{9} + \ldots$"; or you could just take the mean of $\langle 1, 3, 5, 2, 2, 4, 4, 6, 6 \rangle$.

The expected value function $\mathbf{E}$ (also known as "expectation") is a linear function, meaning that for any value c, $\mathbf{E}[c \cdot X] = c \cdot \mathbf{E}[X]$ and $\mathbf{E}[X] + \mathbf{E}[Y] = \mathbf{E}[X + Y]$. Furthermore, $\mathbf{E}[X \cdot Y] = \mathbf{E}[X]\mathbf{E}[Y]$ holds when X and Y are independent. For example, in the random process where we have n red dice and m blue dice, and will roll them and multiply the sum of the red values with the sum of the blue values, the expected value is simply $(3.5 \cdot n) \cdot (3.5 \cdot m) = 12.25 \cdot nm$. For more detail about the linearity of expectation, see the lecture slides.

2.5 Probability Bounds

Theorem 2.1. *(Markov's Inequality)* $P(|X| \geq a) \leq \dfrac{\mathbf{E}[|X|]}{a}$

Proof. Let Y be a random indicator variable that is 1 when $|X| \geq a$ and 0 otherwise.
Note that $\mathbf{E}[Y] = P(|X| \geq a)$.

Trivially, $aY \leq |X|$, because when $|X| < a$ then $Y = 0$ and the left hand side is zero; and when $|X| \geq a$, then $Y = 1$ and the left hand side is a.

Thus, $\mathbf{E}[aY] \leq \mathbf{E}[|X|] \implies a\mathbf{E}[Y] \leq \mathbf{E}[|X|] \implies P(|X| \geq a) \leq \dfrac{\mathbf{E}[|X|]}{a}$ □

The quantity $\mathbf{E}[(X - \mathbf{E}[X])^2]$ is called the *variance* of X, written $\mathrm{Var}(X)$.

Theorem 2.2. *(Chebyshev's Inequality)* $P(|X - \mathbf{E}[X]| \geq d) \leq \dfrac{\mathrm{Var}(X)}{d^2}$

Proof. Apply Markov's inequality to $P((X - \mathbf{E}[X])^2 \geq d^2)$, resulting in $P((X - \mathbf{E}[X])^2 \geq d^2) \leq \dfrac{\mathrm{Var}(X)}{d^2}$.
$(X - \mathbf{E}[X])^2 \geq d^2$ holds if and only if $|X - \mathbf{E}[X]| \geq d$, so $P((X - \mathbf{E}[X])^2 \geq d^2) = P(|X - \mathbf{E}[X]| \geq d)$
Therefore, Chebyshev's Inequality holds. $\square$

Note that $\mathrm{Var}(aX) = \mathbf{E}[(aX - \mathbf{E}[ax])^2] = \mathbf{E}[(aX - a\mathbf{E}[X])^2] = a^2\mathbf{E}[(X - \mathbf{E}[X])^2] = a^2\mathrm{Var}(X)$.

We will use without proof that if X and Y are independent, then $\mathrm{Var}(X + Y) = \mathrm{Var}(X) + \mathrm{Var}(Y)$ (the proof is similar to the proof of linearity of expectation, but messier).

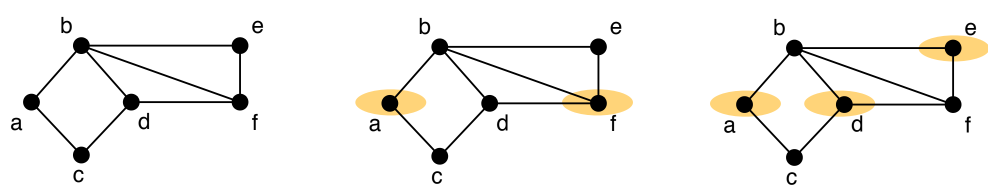
3 Maximal Independent Set (MIS)

In graph theory, an *independent set* is a set of vertices from an undirected graph that have no edges between them. More formally, let a graph $G = (V, E)$ be given. We say that a set $I \subseteq V$ is an independent set if and only if $(I \times I) \cap E = \emptyset$.

For example, if vertices in a graph represent entities and edges represent conflicts between them, an independent set is a group of non-conflicting entities, which is a natural thing to want to know. This turns out to be an important substep in several parallel algorithms since it allows one to find sets of things to do in parallel that don't conflict with each other. For this purpose, it is important to select a large independent set since it will allow more things to run in parallel and presumably reduce the span of the algorithm.

Unfortunately, the problem of finding the overall largest independent set—known as the Maximum Independent Set problem—is **NP**-hard. Its close cousin Maximal Independent Set, however, admits efficient algorithms and is a useful approximation to the harder problem.

More formally, the *Maximal Independent Set* (MIS) problem is: given an undirected graph $G = (V, E)$, find an independent set $I \subseteq V$ such that for all $v \in (V \setminus I)$, $I \cup \{v\}$ is not an independent set. Such a set I is maximal in the sense that we can't add any other vertex and keep it independent, but it easily may not be a maximum—i.e. largest—independent set in the graph.



maximal independent set

maximal and maximum independent set

For example, in the graph above, the set $\{a, d\}$ is an independent set, but not maximal because $\{a, d, e\}$ is also an independent set. On the other hand, the set $\{a, f\}$ is a maximal independent set because there's no vertex that we can add without losing independence. Note that in MIS, we are *not* interested in computing the overall-largest independent set: while maximum independent sets are maximal independent sets, maximal independent set are not necessarily maximum independent sets! Staying with the example above, $\{a, f\}$ is a maximal independent set but not a maximum independent set because $\{a, d, e\}$ is independent and larger.

3.1 Sequential MIS

Let's first think about how we would compute an MIS if we don't care about parallelism. We will start by thinking about the effect of picking a vertex v as part of our independent set I. By adding v to I, we know that

none of v's neighbors can be added to I. This motivates an algorithm that picks an arbitrary vertex v from G, add v to I, and derive G' from G such that each vertex of G' is independent of I and can be picked in the next step. This means that G' contains all vertices other than v and its neighbors. In the terminology of last lecture, G' is the induced subgraph on $V \setminus (N(v) \cup \{v\})$. We have the following algorithm:

```
function seqMIS((V,E),I) =
if |V| = 0 then I else
  let
     v = pickAnyOne(V)
     V' = V \ (N(v) ∪ {v})
     E' = E ∩ (V' × V')
  in seqMIS((V',E'), I ∪ {v})
  end
```

In words, the algorithm proceeds in iterations until the whole graph is exhausted. Each iteration involves picking an arbitrary vertex, which is added to the independent set I, and removing the vertices v, together with v's neighbors $N(v)$, and edges incident on these vertices. Thus, each round picks a new vertex and removes exactly the vertices that can no longer be added to I, and nothing more. It is not difficult to convince ourselves that this algorithm indeed computes a maximal independent set of G. With a proper implementation (e.g., using arrays), this algorithm takes $O(m+n)$ work.

Q: How do we know this will really compute an MIS?
A: I is really an independent set because, at every step of the algorithm, there are no edges between any vertex in V and any vertex in I, so no vertex in I is a neighbor of an edge added before it, or after it. To show that I is maximal, suppose we can add a vertex v to the returned set I. Then there is no $u \in I$ such that $v \in N(u)$. But then v should never have been removed from V and it would eventually be picked and added to I.

Some of the operations in each round of this algorithm can be made parallel. However, the rounds must still be run in sequence, adding one vertex to the MIS at a time. We want to run multiple rounds in parallel.

3.2 Parallel MIS

The first attempt to parallelize the above algorithm might be to pick several vertices at once, add all of them to the MIS at once, and update the graph in parallel.

Q: What's the problem with this?
A: We may choose two vertices that are neighbors, and can't both be added.

Q: Where have we seen a similar problem before?
A: (For example) star contraction. We needed to make sure that the same vertex wasn't added to two stars.

To solve the problem in star contraction, we flipped coins and assigned each vertex to either be a center or a satellite. This resolved the abiguity caused by symmetry. We want to do something similar.

Attempt 1: Flip a coin for every vertex. Only choose a vertex if it comes up heads.
Problem: Two neighbors could both come up heads, and both be chosen.

Attempt 2: Only choose a vertex if it's heads and all its neighbors are tails.
Problem: As we saw with edge contraction, the probability of choosing a vertex v is $\frac{1}{2} \cdot \frac{1}{2^{|N(v)|}}$. If the graph is highly connected, there's a good chance that we won't choose any vertex in a particular round.

How can we make sure we choose at least one vertex on each round, but never choose two neighbors? One idea is to choose a random number in $[0, |V|)$ for each vertex instead of flipping a coin. We choose a vertex if it is a *local maximum*, that is, if its number is greater than those of all of its neighbors. By definition, if a vertex is

a local maximum, none of its neighbors are, and there is likely to be at least one local maximum. If we ensure that all of the random numbers are distinct, there is guaranteed to be at least one local maximum (the global maximum).

We modify the pseudocode of the sequential algorithm to do this:

function $\texttt{parMIS}((V, E), I) =$
if $|V| = 0$ **then** I **else**
  let
    $H = \{v \mapsto \texttt{hash}(v) : v \in V\}$
    $U = \{v \in V \mid H[v] > \max_{u \in N(v)} H[u]\}$
    $V' = V \setminus (U \cup \bigcup_{u \in U} N(u))$
    $E' = E \cap (V' \times V')$
  in $\texttt{parMIS}((V', E'), I \cup U)$
  end