Getting to know each other

While people are coming in, please turn to a neighbor and introduce yourself. Get to know a little about them... potential major, hometown, class year, what's in their playlist, etc.

COSC 290 Discrete Structures

Lecture 1: Basic Data Types

Prof. Michael Hay

Wednesday, Jan. 24, 2018

Colgate University

Plan for today

1. Logistics
2. Basic data types: booleans, numbers, common operations
3. Real-world application: frequent itemset mining

Logistics

Logistics

- Department "tea": tomorrow at 11:30am in research lounge (glass door at end of hall). Free food, get to know other CS majors/minors
- Next Tuesday (11:30am): department tea to hear about summer research projects
- Problem set 0!
- Peer-led team-based workshops: strongly encouraged to sign up for a section (see moodle)!
- No lab today

Reading

Do reading *before* class.

Focus on *main* ideas (this first chapter is a lot of little ideas/concepts).

Read "Computer Science Connections"

Check out "Chapter At A Glance" (last section of each chapter)

Basic data types: booleans, numbers, common operations

Booleans, numbers, operations

- Booleans $= \{True, False\}$
- Integers $\mathbb{Z}$
- Reals $\mathbb{R}$, non-negative reals $\mathbb{R}^{\geq 0}$
- Rationals $\mathbb{Q}$
- Absolute value $|x|$, floor $\lfloor x \rfloor$, ceiling $\lceil x \rceil$
- Logarithms and exponentials
- Modulus: if $x \bmod 2 = 0$, then x is...?
- Summations $\sum_{i=1}^{n} x_i$ and products $\prod_{i=1}^{n} x_i$

Polling questions

Rules of the game.

- *(Before class, you prepare yourself by reading the textbook and completing any problem sets)*
- I ask a question.
- You first answer it by yourself... no talking!
- Then discuss in groups of 3-4 students.
- Answer the question a second time.
- I will ask someone to answer and we will discuss.

Why?

Poll everywhere

On a device of your choice, go to `pollev.com/cosc290`

Floors, Ceilings

Definition (Floor)
The floor of a real number x, written $\lfloor x \rfloor$, denotes the *largest* integer that is *less* than or equal to x.

Definition (Ceiling)
The ceiling of a real number x, written $\lceil x \rceil$, denotes the *smallest* integer that is *greater* than or equal to x.

Examples:

- $\lfloor 5.3234 \rfloor = 5$
- $\lceil 5.3234 \rceil = 6$
- $\lfloor 12.0 \rfloor = 12$
- $\lceil 12.0 \rceil = 12$

Poll: floors and ceilings

Consider the following quantity,

$$x - \frac{\lfloor x \rfloor + \lceil x \rceil}{2}$$

For what value(s) of x in the interval $[2, 3]$ is this quantity the *largest*?

1. 2
2. 2.000...1
3. 2.5
4. 2.99999999
5. 3
6. More than one of the above

Vote by going to this site: `pollev.com/cosc290`

Summations

Quick review of notation:

$$\sum_{i=1}^{n} x_i \coloneqq x_1 + x_2 + \cdots + x_n$$

Example:

$$\begin{aligned}\sum_{i=-2}^{3} i^2 &= (-2)^2 + (-1)^2 + (0)^2 + 1^2 + 2^2 + 3^2 \\ &= 4 + 1 + 0 + 1 + 4 + 9 \\ &= 19\end{aligned}$$

Poll: summations

Calculate this summation,

$$\sum_{i=1}^{3} \sum_{j=1}^{3} (i \cdot j)$$

1. 6
2. 10
3. 18
4. 24
5. 36

Vote by going to this site: `pollev.com/cosc290`

Poll: summation identity

Consider the following equation,

$$\sum_{i=1}^{n} 2 \cdot i = \sum_{j=2}^{m} \left\lfloor \frac{j}{2} \right\rfloor$$

What value of m makes the right-hand side equal the left-hand side?

1. $m = n/2$
2. $m = n$
3. $m = 2n$
4. $m = 2n + 1$
5. None of the above

Vote by going to this site: `pollev.com/cosc290`

Real-world application: frequent itemset mining

CS connections & Real-world applications

Book chapters have short sections called "Computer Science Connections." Please consider these as part of the assigned reading.

Labs (and sometimes class) will touch on real-world applications.

Why?

- help reinforce your understanding of concepts
- help you see value in learning these concepts

Frequent Itemset Mining

Your internship at @WalmartLabs: analyze data on customer purchases.

Specifically, find all frequent itemsets. A frequent itemset is a collection of items that are frequently purchased together (by at least 1% of customers, for example).

Why might this be useful?

Input

Data on consumer purchases.

Representation: A list of n transactions $t_1, \ldots, t_n$ where each transaction t_i is represented as a *set* of items purchased.

Example:

$$t_1 = \{ \text{soy milk}, \text{coffee} \}$$
$$t_2 = \{ \text{milk}, \text{orange juice}, \text{cocoa puffs} \}$$
$$\ldots$$
$$t_n = \{ \text{organic tofu}, \text{broccoli}, \text{coffee}, \text{soy milk} \}$$

Why represent a transaction as a *set* of items? What information do we lose with this representation?

Support for an itemset

Suppose we have a particular itemset in mind, say $c := \{ \text{coffee}, \text{soy milk} \}$.

We want to know the support for the itemset: the number of transactions in which the items in c were purchased together.

Example: suppose we have $n = 5$ transactions.

$$t_1 = \{ \text{soy milk}, \text{coffee} \}$$
$$t_2 = \{ \text{milk}, \text{orange juice}, \text{cocoa puffs} \}$$
$$t_3 = \{ \text{soy milk}, \text{sugar}, \text{coffee} \}$$
$$t_4 = \{ \text{organic tofu}, \text{broccoli}, \text{coffee}, \text{soy milk} \}$$
$$t_5 = \{ \text{coffee}, \text{orange juice} \}$$

The support for c is... 3, because it occurs in three transactions (t_1, t_3, and t_4).

Poll: support for an itemset

The support of an itemset c is the number of transactions in which the items in c were purchased together. Given these 5 transactions,

$t_1 = \{ \text{soy milk}, \text{coffee} \}$
$t_2 = \{ \text{milk}, \text{orange juice}, \text{cocoa puffs} \}$
$t_3 = \{ \text{soy milk}, \text{sugar}, \text{coffee} \}$
$t_4 = \{ \text{organic tofu}, \text{broccoli}, \text{coffee}, \text{soy milk} \}$
$t_5 = \{ \text{coffee}, \text{orange juice} \}$

What is the support of itemset $c := \{ \text{orange juice}, \text{soy milk} \}$?

1. 0
2. 2
3. 3
4. 5

Finding frequent itemsets

An itemset is a frequent itemset if its support is above some threshold (e.g., 1% of all transactions).

The data mining task is to (efficiently) find all frequent itemsets.

Input: A list of transactions $T = t_1, \ldots, t_N$
Output: Prints out all frequent itemsets.

```
1: Let U be the set of all items that occur in transactions T.
2: for all itemsets c you can make from items in U do
3:     support := 0                          ▷ for each c, compute support
4:     for all t_i ∈ T do
5:         if transaction t_i includes c then
6:             support := support + 1                ▷ increment support
7:     if support is above threshold then        ▷ itemset c is frequent
8:         print c
```

Example input: $T = t_1, t_2, \ldots, t_5$

$t_1 = \{ \text{soy milk}, \text{coffee} \}$
$t_2 = \{ \text{milk}, \text{orange juice}, \text{cocoa puffs} \}$
$t_3 = \{ \text{soy milk}, \text{sugar}, \text{coffee} \}$
$t_4 = \{ \text{organic tofu}, \text{broccoli}, \text{coffee}, \text{soy milk} \}$
$t_5 = \{ \text{coffee}, \text{orange juice} \}$

Efficiency considerations

In coming lectures and labs, we will...

- ... look at this problem a little more formally, using set notation and set operations
- ... think about how to calculate frequent itemsets *efficiently*
- ... implement a frequent itemset miner in Java

Frequent itemset mining in the real world

Fun fact: Walmart found that when a hurricane/storm is forecast, people stock up on...

- Bottled water
- Flashlights
- Batteries
- Pop tarts
- Beer

Source: `https://tinyurl.com/43zxg8z`