Entropy and decisions
Overview

• This lecture: Information theory.
  • Entropy.
  • Mutual information, etc.
• Next lecture: Decisions.
  • Hypothesis testing.
  • T tests.
  • Multiple comparisons.
Information

• Imagine Darth Vader is about to say either “yes” or “no” with **equal** probability.
  • You don’t know what he’ll say.

• You have a certain amount of **uncertainty** – a lack of information.
Information

• Imagine you then observe Darth Vader saying “no”
• Your uncertainty is gone; you’ve received information.
• How much information do you receive about event $x$ when you observe it?

"Choosing 1 out of 2" gives a bit of information

$$I(x) = 1 \text{ bit for } P(x) = \frac{1}{2}$$
Information

- Imagine there is both Darth Vader and Varth Dader.
- Observing what both DV and VD say gives us 2 bits of information.
- There are $2^2$ scenarios with equal possibilities:
  - Yes/Yes, Yes/No, No/Yes, No/No
Information

• So $I(x)$=2 bits is brought by $P(x) = \frac{1}{2^2}$

• $I(x)$ doubles when $\frac{1}{P(x)}$ is squared.

• Let’s describe $I(x)$ with negative log likelihood:

$$I(x) = \log_2 \frac{1}{P(x)}$$

Going back to the “yes/no” example:

$$I(no) = \log_2 \frac{1}{P(no)} = \log_2 \frac{1}{1/2} = 1 \text{ bit}$$

Note 1: Negative log likelihood is also called surprisal.

Note 2: information contents computed with log base 2 has unit “bit”. Log base e => unit “nat”.

For capturing the Logarithm relationship

$I(x) = -\log_2 P(x)$;
So here comes the negation
Information

- Imagine Darth Vader is about to roll a fair die.
- You have more uncertainty about an event because there are more possibilities.
- You receive more information when you observe it.

\[ I(x) = \log_2 \frac{1}{P(6)} \]
\[ = \log_2 \frac{1}{\frac{1}{6}} \approx 2.58 \text{ bits} \]
Information can be additive

• One property of $I(x) = \log_2 \frac{1}{P(x)}$ is additivity.

• From $k$ independent events $x_1 \ldots x_k$:
  • Does $I(x_1 \ldots x_k) = I(x_1) + I(x_2) + \ldots + I(x_k)$?
  • The answer is yes!

\[
I(x_1 \ldots x_k) = \log_2 \frac{1}{P(x_1 \ldots x_k)}
\]

\[
= \log_2 \frac{1}{P(x_1) \ldots P(x_k)} = \log_2 \frac{1}{P(x_1)} + \ldots + \log_2 \frac{1}{P(x_k)}
\]

\[
= I(x_1) + I(x_2) + \ldots + I(x_k)
\]
Aside: Information in computers

• The unit bit appears familiar to the units describing file sizes...
• And they are related!
• $1 \text{ GB} = 2^{10} \text{ MB} = 2^{20} \text{ KB} = 2^{30} \text{ Bytes}$, where:
  • 1 Byte = 8 bits.
  • Historically: 1 byte was used to store one character.
• File sizes in computers are described by the amount of information.
  • The file sizes also depend on the method of encoding (approx. “file format”)

CSC401/2511 – Winter 2023
Events and random variables

• An event $x$ is a sample from a random variable $X$.
• Example 1:
  • $X$: Darth Vader saying something (either yes or no)
  • $x$: What DV says ($x = “no”$)
• Example 2:
  • $X$: Darth Vader rolling a die
  • $x$: The side facing upwards (e.g., $x = 3$)
• $x$ is deterministic. $X$ is random.
• $x$ is the output emitted by the “source” $X$. 
Information with unequal events

- The random variable $X$ can take possible values: 
  $\{v_1, v_2, \ldots, v_n\}$.
- Each value has its own probability $\{p_1, p_2, \ldots, p_n\}$

- What is the average amount of information we get in observing the output of $X$?

- You still have 6 events that are possible – but you’re fairly sure it will be ‘No’.
Entropy

- **Entropy**:  \( n. \) the **expected** information gaining from observing the events of the random variable \( X \).

\[
H(X) = E_x[I(x)] = \sum_x p(x) \log \frac{1}{p(x)}
\]

**Notes:**
1. Entropy is defined towards a random variable.
2. Entropy is the average uncertainty inherent in a random variable.
Entropy – examples

\[ H(X) = \sum_i p_i \log_2 \frac{1}{p_i} \]

\[ = 0.7 \log_2(1/0.7) + 0.1 \log_2(1/0.1) + \cdots \]

\[ = 1.542 \text{ bits} \]

There is **less** average uncertainty when the probabilities are ‘skewed’.

\[ H(X) = \sum_i p_i \log_2 \frac{1}{p_i} = 6 \left(\frac{1}{6} \log_2 \frac{1}{1/6}\right) \]

\[ = 2.585 \text{ bits} \]
Entropy characterizes the distribution

- ‘Flatter’ distributions have a **higher** entropy because the choices are **more equivalent**, on average.
- So which of these distributions has a **lower** entropy?
Low entropy makes decisions easier

• When predicting the next event, we’d like a distribution with lower entropy.
• Low entropy ≡ less uncertainty
Bounds on entropy

- **Maximum**: uniform distribution $X_1$. Given $M$ choices,

$$H(X_1) = \sum_i p_i \log_2 \frac{1}{p_i} = \sum_i \frac{1}{M} \log_2 \frac{1}{1/M} = \log_2 M$$

- **Minimum**: only one choice, $H(X_2) = p_i \log_2 \frac{1}{p_i} = 1 \log_2 1 = 0$
Alternative notions of entropy

- Entropy is equivalently:
  - The average amount of information provided by an observation of a random variable,
  - The average amount of uncertainty you have before an observation of a random variable,
  - The average amount of ‘surprise’ you receive during the observation,
  - The number of bits needed to communicate that random variable
    - Aside: Shannon showed that you cannot have a coding scheme that can communicate it more efficiently than $H(S)$
Some terms

- Joint entropy
- Conditional entropy
- Mutual information
- Cross entropy
Entropy of several variables

- Consider the vocabulary of a meteorologist describing **Temperature** and **Wetness**.
  - **Temperature** = \{hot, mild, cold\}
  - **Wetness** = \{dry, wet\}

$$P(W = \text{dry}) = 0.6,$$
$$P(W = \text{wet}) = 0.4$$

$$P(T = \text{hot}) = 0.3,$$
$$P(T = \text{mild}) = 0.5,$$
$$P(T = \text{cold}) = 0.2$$

\[ H(W) = 0.6 \log_2 \frac{1}{0.6} + 0.4 \log_2 \frac{1}{0.4} = 0.970951 \text{ bits} \]

\[ H(T) = 0.3 \log_2 \frac{1}{0.3} + 0.5 \log_2 \frac{1}{0.5} + 0.2 \log_2 \frac{1}{0.2} = 1.48548 \text{ bits} \]

But \( W \) and \( T \) are not independent, \( P(W, T) \neq P(W)P(T) \)
Joint entropy

• **Joint Entropy:** *n.* the average amount of information needed to specify multiple variables *simultaneously*.

\[
H(X, Y) = \sum_x \sum_y p(x, y) \log_2 \frac{1}{p(x, y)}
\]

• **Hint:** this is *very* similar to univariate entropy – we just replace univariate probabilities with joint probabilities and sum over everything.
Entropy of several variables

- Consider joint probability, $P(W, T)$

<table>
<thead>
<tr>
<th></th>
<th>cold</th>
<th>mild</th>
<th>hot</th>
</tr>
</thead>
<tbody>
<tr>
<td>dry</td>
<td>0.1</td>
<td>0.4</td>
<td>0.1</td>
</tr>
<tr>
<td>wet</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>0.3</td>
<td>0.5</td>
<td>0.2</td>
</tr>
</tbody>
</table>

- Joint entropy, $H(W, T)$, computed as a sum over the space of joint events $(W = w, T = t)$

$$H(W, T) = 0.1 \log_2 \frac{1}{0.1} + 0.4 \log_2 \frac{1}{0.4} + 0.1 \log_2 \frac{1}{0.1} + 0.2 \log_2 \frac{1}{0.2} + 0.1 \log_2 \frac{1}{0.1} + 0.1 \log_2 \frac{1}{0.1} = 2.32193 \text{ bits}$$

Notice $H(W, T) \approx 2.32 < 2.46 \approx H(W) + H(T)$
Entropy given knowledge

• In our example, **joint entropy** of two variables together is lower than the sum of their individual entropies
  • \( H(W, T) \approx 2.32 < 2.46 \approx H(W) + H(T) \)

• Why?
  
  • Information is **shared** among variables
    • There are **dependencies**, e.g., between temperature and wetness.
    • E.g., if we knew **exactly** how wet it is, is there **less confusion** about what the temperature is ... ?
Conditional entropy

• **Conditional entropy**: \( n. \) the **average** amount of information needed to specify one variable given that you know another.

\[
H(Y|X) = \sum_{x \in X} p(x)H(Y|X = x)
\]

• **Comment**: this is the expectation of \( H(Y|X) \), w.r.t. \( x \).
Entropy given knowledge

- Consider **conditional** probability, $P(T|W)$

\[
P(T|W) = \frac{P(W,T)}{P(W)}
\]

<table>
<thead>
<tr>
<th>$P(W,T)$</th>
<th>$T = \text{cold}$</th>
<th>mild</th>
<th>hot</th>
<th>$P(W)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W = \text{dry}$</td>
<td>0.1</td>
<td>0.4</td>
<td>0.1</td>
<td>0.6</td>
</tr>
<tr>
<td>wet</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>0.3</td>
<td>0.5</td>
<td>0.2</td>
<td>1.0</td>
</tr>
</tbody>
</table>

| $P(T | W)$ | $T = \text{cold}$ | mild | hot | $P(W)$ |
|---------|------------------|------|-----|---------|
| $W = \text{dry}$ | 0.1/0.6 | 0.4/0.6 | 0.1/0.6 | 1.0 |
| wet     | 0.2/0.4 | 0.1/0.4 | 0.1/0.4 | 1.0 |
Entropy given knowledge

• Consider **conditional** probability, \( P(T|W) \)

<table>
<thead>
<tr>
<th>( P(T \mid W) )</th>
<th>( T = \text{cold} )</th>
<th>( \text{mild} )</th>
<th>( \text{hot} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W = \text{dry} )</td>
<td>1/6</td>
<td>2/3</td>
<td>1/6</td>
</tr>
<tr>
<td>( \text{wet} )</td>
<td>1/2</td>
<td>1/4</td>
<td>1/4</td>
</tr>
</tbody>
</table>

\[
H(T \mid W = \text{dry}) = H \left( \left\{ \frac{1}{6}, \frac{2}{3}, \frac{1}{6} \right\} \right) = 1.25163 \text{ bits}
\]

\[
H(T \mid W = \text{wet}) = H \left( \left\{ \frac{1}{2}, \frac{1}{4}, \frac{1}{4} \right\} \right) = 1.5 \text{ bits}
\]

• **Conditional entropy** combines these:

\[
H(T \mid W) = 0.6 [p(W = \text{dry})H(T \mid W = \text{dry})] + 0.4 [p(W = \text{wet})H(T \mid W = \text{wet})] = 1.350978 \text{ bits}
\]
Equivocation removes uncertainty

- Remember $H(T) = 1.48548$ bits
- $H(W, T) = 2.32193$ bits
- $H(T|W) = 1.350978$ bits

- How much does $W$ tell us about $T$?
  - $H(T) - H(T|W) = 1.48548 - 1.350978 \approx 0.1345$ bits
  - Well, a little bit!

Entropy (i.e., confusion) about temperature is reduced if we know how wet it is outside.
Perhaps $T$ is more informative?

- Consider **another** conditional probability, $P(W|T)$

| $P(W|T)$ | $T = \text{cold}$ | $T = \text{mild}$ | $T = \text{hot}$ |
|---------|------------------|------------------|------------------|
| $W = \text{dry}$ | 0.1/0.3 | 0.4/0.5 | 0.1/0.2 |
| wet | 0.2/0.3 | 0.1/0.5 | 0.1/0.2 |
| 1.0 | 1.0 | 1.0 |

- $H(W|T = \text{cold}) = H\left(\left\{\frac{1}{3}, \frac{2}{3}\right\}\right) = 0.918295$ bits
- $H(W|T = \text{mild}) = H\left(\left\{\frac{4}{5}, \frac{1}{5}\right\}\right) = 0.721928$ bits
- $H(W|T = \text{hot}) = H\left(\left\{\frac{1}{2}, \frac{1}{2}\right\}\right) = 1$ bit
- $H(W|T) = 0.8364528$ bits
Equivocation removes uncertainty

• $H(T) = 1.48548$ bits
• $H(W) = 0.970951$ bits
• $H(W, T) = 2.32193$ bits
• $H(T|W) = 1.350978$ bits
• $H(T) - H(T|W) \approx 0.1345$ bits

• How much does $T$ tell us about $W$ on average?
  • $H(W) - H(W|T) = 0.970951 - 0.8364528 \approx 0.1345$ bits

• Interesting ... is that a coincidence?
Mutual information

- **Mutual information**: *n.* the **average** amount of information **shared** between variables.

\[
I(X; Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)
\]

\[
= \sum_{x,y} p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)}
\]

- **Hint**: The amount of uncertainty **removed** in variable \(X\) if you know \(Y\).
- **Hint2**: If \(X\) and \(Y\) are **independent**, \(p(x, y) = p(x)p(y)\), then

\[
\log_2 \frac{p(x, y)}{p(x)p(y)} = \log_2 1 = 0 \ \forall x, y – \text{there is no mutual information!}
\]
Relations between entropies

\[ H(X, Y) = H(X) + H(Y) - I(X; Y) \]
Aside: Kullback-Leibler divergence

• **KL divergence** measures the **dis-similarity** between two probability distributions $p(X)$ and $q(X)$:

$$KL(p||q) = \sum_X p(X) \log \frac{p(X)}{q(X)}$$

• $KL \geq 0$, with equality reached at $p(X) = q(X)$.

• KL is asymmetrical: $KL(p||q) \neq KL(q||p)$.

• Usually, it’s hard to precisely know the KL divergence of two distributions.

• KL is frequently used in reinforcement learning.
Aside: Cross-entropy

- **Cross-entropy** also measures the dis-similarity between two distributions.
- It is used to measure the quality of a predicted distribution $q(Y|X)$ with respect to the ground truth $p(Y|X)$:

$$H(p, q) = \frac{1}{N} \sum_{x} p(Y|X) \log \frac{1}{q(Y|X)}$$

- Cross-entropy is frequently used in machine learning as the target for optimization, i.e., cross-entropy loss.
  - More details in CSC311.
  - But ML uses cross-entropy in base e.
Lecture review questions

By the end of this lecture, you should be able to:

• Describe random variable and random events.
• Compute entropy, joint entropy, conditional entropy, and mutual information.

• (Not on exam) Be familiar with the terms KL divergence and cross entropy.

• Anonymous feedback form: https://forms.gle/W3i6AHaE4uRx2FAJA
Decisions
Does the sun rise from the east?

- Why are you sure it’s the east, but not the west?
- Because there are repeated observations!
How does new knowledge occur?

Knowledge: “The sun rises from the east.”
The knowledge comes from repeated observations of the “sun rise” event.

There is a “hypothesize – confirm” workflow in discovering new knowledge from the observations.

Hypothesis testing is a *standardized* procedure of this type of discovery.
Procedure of a statistical test

Step 1: **State** a hypothesis.
- Null hypothesis $H_0$ and alternative hypothesis $H_1$; more in the next slide.

Step 2: **Compute** some test statistics.
- For example: p-value

Step 3: **Compare** the statistics to a critical value and report the test results.
- E.g., compare $p$ to $\alpha = 0.05$ ("significance level"). If $p < 0.05$, reject $H_0$. Otherwise, do not reject $H_0$. 
Null and Alternative Hypotheses

• **Null hypothesis** $H_0$ usually states that “nothing has changed”.

• **Alternative hypothesis** $H_1$ usually states that “there are some meaningful findings”.
**t tests**

*t-test* is very frequently used in NLP. Some problems that can be studied by *t*-test include:

Q1: Does Elon send tweets of 100 words long?

Q2: I added a layer to the neural network. Is the prediction accuracy better than the baseline?

Q3: A group of participants try a recipe for a month. Do their weights change?
Sample vs. population

Samples are known to you, but the population is not.

Q1: Does Elon send tweets of 100 words long?

A sample is an event “observe a tweet”.

The population is the random variable “Elon sends tweets”.

Recall: “Darth Vader saying something vs. what DV says”
One-sample \( t \) test

Does Elon send tweets of 100 words long?

\( H_1: \) The population mean is different from 100.

\( H_0: \) Otherwise. There’s no new finding here.

Compare the sample mean (average tweet length of a sample of e.g., \( N=50 \) Elon tweets) with the hypothetical population mean (\( \mu = 100 \)).

```python
from scipy.stats import ttest_1samp
t, p = ttest_1samp(lengths, popmean=100)
```

Note 1: The true population mean is unknown.
One-sample $t$ test

from scipy.stats import ttest_1samp
t, p = ttest_1samp(lengths, popmean=100)

• Note 2: We need to assume the population is **normally** distributed.
  • You can double check by **Shapiro-Wilks test** (also in scipy.stats package)
  • If the population does not follow normal distribution, use **Mann-Whitney U test** instead.
  • As an exploratory analysis, just do a **quantile-quantile plot** (**Q-Q plot**) against a normal distribution.
One-sample $t$ test

```python
from scipy.stats import ttest_1samp
t, p = ttest_1samp(lengths, popmean=100)
```

- Note 3: The $p$ value means, *approximately*, how likely is the sample mean equal 100.
  - $p<0.05$: reject the null hypothesis $H_0$.
  - Otherwise: we don’t have sufficient evidence to reject $H_0$.

- Note 4: The *degree-of-freedom* equals $N - 1$.
  - For details, please refer to a statistics course.
One-sample \( t \) test

- Note 5: The alternative hypotheses can differ. \( H_0 \) means “otherwise” in all cases.

\[
t, p = \texttt{ttest\_1samp(lengths, popmean=100, alternative=“two-sided”)}
\]

- \( H_1 \): the population mean is \textit{different} from 100.
- Two-sided \( t \)-test is the default in \texttt{ttest\_1samp}.

\[
t, p = \texttt{ttest\_1samp(lengths, popmean=100, alternative=“greater”)}
\]

- \( H_1 \): the population mean is \textit{greater than} 100.

\[
t, p = \texttt{ttest\_1samp(lengths, popmean=100, alternative=“less”)}
\]

- \( H_1 \): the population mean is \textit{less than} 100.
Two-sample $t$ test

I added a layer to the neural network. Is the prediction accuracy better than the baseline?

$H_1$: Yes. The new configuration has higher accuracy.

$H_0$: Otherwise. There’s no new finding here.

Collect the samples (accuracy from $N_1$ experiments) using the new and $N_2$ from the old configuration.

```python
from scipy.stats import ttest_ind
t, p = ttest_ind(old_results, new_results)
```
Two-sample $t$ test

from scipy.stats import ttest_ind

t, p = ttest_ind(old_results, new_results)

- Note 1: The population means of both populations are unknown.
- Note 2: The two populations should be independent.
- Note 3: The p value means, approximately, how likely the two population means are equal.
  - $p < 0.05$: reject $H_0$
- Note 4: The degree-of-freedom equals $N_1 + N_2 - 2$
Paired $t$ test

A group of $N$ participants try a recipe for a month. Do their weights change?

$H_1$: Yes. This recipe changes the weights.
$H_0$: Otherwise.

Collect the participants’ weights before and after the month, and plug in the formula:

```python
from scipy.stats import ttest_rel
t, p = ttest_rel(before_weights, after_weights)
```
Paired $t$ test

from scipy.stats import ttest_rel
t, p = ttest_rel(before_weights, after_weights)

• Note 1: The degree of freedom is $N - 1$
• Note 2: Paired t-test is equivalent to one-sample t test of the weight differences against 0.
• Note 3: The p value means, *approximately*, how likely the difference is 0.
  • $p < 0.05$: reject $H_0$.
• Note 4: If we incorrectly use two-sample t test when there are obvious one-to-one correspondence between groups, then the p values could be *inflated*.
Summary: Types of $t$-tests

- **One-sample $t$-test**: whether the population mean equals $\mu$.
  - Population mean $X$ is a random variable.
  - `scipy.stats.ttest_1samp`

- **Two-sample $t$-test**: whether the mean of two populations, $X$ and $Y$, equal each other.
  - `scipy.stats.ttest_ind`

- **Paired $t$-test**: whether $X - Y$ equals a known value $\mu$.
  - `scipy.stats.ttest_rel`
Multiple comparisons

• Imagine you’re flipping a coin to see if it’s fair. You claim that if you get ‘heads’ in 9/10 flips, it’s biased.

• Assuming $H_0$, the coin is fair, the probability that a fair coin would come up heads $\geq 9$ out of 10 times (i.e., appear biased) is:

$$\frac{(10 + 1) \times 0.5^{10}}{10!} = 0.0107$$

Number of ways 9 flips are heads    Number of ways all 10 flips are heads
Multiple comparisons

• But imagine that you’re simultaneously testing 173 coins – you’re doing 173 (multiple) comparisons.
• If you want to see if a specific chosen coin is fair, you still have only a 1.07% chance that it will appear biased.
• But if you don’t preselect a coin, what is the probability that none of these fair coins will accidentally appear biased?
  
  \[(1 - 0.0107)^{173} \approx 0.156\]

• If you’re testing 1000 coins?
  
  \[(1 - 0.0107)^{1000} \approx 0.0000213\]
Multiple comparisons

• The more tests you conduct with a statistical test, the more likely you are to accidentally find spurious (incorrect) significance accidentally.

• **Bonferroni correction** is an adjustment method:
  • Divide your level of significance required $\alpha$, by the number of comparisons.
  • E.g., if $\alpha = 0.05$, and you’re doing 173 comparisons, each would need $p < \frac{0.05}{173} \approx 0.00029$ to be considered significant.
P-hacking

• Once you get a result, do **not** do any of the following to try to increase the significance:
  • Re-sample the data.
  • Change one-tailed test to two-tailed tests.
  • Change the type of tests and pick a significant one.
  • ...
  • These are called “**p-hacking**”.

• The harm of p-hacking? “Discovery of false knowledge”.

• If a statistical test leads to insignificant results, just say ”the result is not significant”.
  • Perhaps also report the p value in your report.
Lecture review questions

By the end of this lecture, you should be able to:

• Describe statistical tests.
• Describe and carry out t-test.
  • Check data for the assumptions of t-tests.
  • Identify different types of t-tests and know when to use which test.
• Describe the multiple comparison problem and adjust for the problem.
• Be familiar with p-hacking, and its harm.

• Anonymous feedback form: [https://forms.gle/W3i6AHaE4uRx2FAJA](https://forms.gle/W3i6AHaE4uRx2FAJA)