Combinatorics
Chris Piech
CS109, Stanford University
Review
cs109.stanford.edu
CS109 Community

Dedicated, intelligent, hardworking teaching assistants
We are counting:

# of **events**, 
# of **outcomes**, 
# of **objects**
Two Key Rules

Counting outcomes with **or:**

*Inclusion Exclusion:*

If outcomes can come from set A **or** set B, then the total number of outcomes is \(|A| + |B| - |A \cap B|\).

Counting outcomes with **steps:**

*Product Rule of Counting:*

If outcomes are generated via a process with \(r\) steps, where step \(i\) has \(n_i\) outcomes, then the total number of outcomes is:

\[
n_1 \times n_2 \times \cdots \times n_r = \prod_{i=1}^{r} n_i
\]
How Many Unique 6 digit passcodes?

Approach: count by steps

**Step 1**: first digit in passcode  
(10 outcomes)

**Step 2**: second digit in passcode  
(10 outcomes)

...  

**Step 6**: second digit in passcode  
(10 outcomes)

\[
total = n_1 \times n_2 \times \cdots \times n_r
\]

\[
= 10 \times 10 \times 10 \times 10 \times 10 \times 10
\]
End Review
Counting tasks on $n$ objects

- Sort objects (permutations)
- Choose $k$ objects (combinations)
- Put objects in $r$ buckets
Combinatorics

Counting tasks on \( n \) objects

- Sort objects (permutations)
  - Distinct: \( n! \)
  - Some Distinct: \( \frac{n!}{n_1!n_2!\ldots} \)

- Choose \( k \) objects (combinations)
  - Distinct: \( \binom{n}{k} = \frac{n!}{k!(n-k)!} \)

- Put objects in \( r \) buckets
  - Distinct: \( r^n \)
  - None Distinct: \( \frac{(n + r - 1)!}{n!(r-1)!} \)
Combinatorics

Counting tasks on \( n \) objects

- Sort objects (permutations)
- Choose \( k \) objects (combinations)
- Put objects in \( r \) buckets

Distinct
Sort $n$ Distinct Objects

Ayesha  Tim  Irina  Joey  Waddie
Sort $n$ Distinct Objects

Sort 5 distinct cans:

Step 1: Chose first can (5 options)

Irina
Sort $n$ Distinct Objects

Sort 5 distinct cans:

Step 1: Chose first can (5 options)

Step 2: Chose second can (4 options)

$5 \times 4 \times 3 \times 2 \times 1 = 120$ unique sorts

Irina

Waddie
Def Permutations:

A permutation is an ordered arrangement of distinct object. 

*n* objects can be permuted in:

\[ n \times (n - 1) \times (n - 2) \times \cdots \times 2 \times 1 = n! \]

(Select 1st object out of *n*, then 2nd object out of *n* − 1, etc.)
Sort Distinct Objects

Ayesha  Tim  Irina  Joey  Waddie

= 120
Sort Semi-Distinct Objects

Coke  Tim  Coke  Joey  Waddie

= 120/2
Making perms of distinct objects is a two step process.

perms of distinct objects = perms considering some objects are indistinct \times perms of just the indistinct objects

Step 1

Step 2
Sort Semi-Distinct Objects

perms of distinct objects = perms considering some objects are indistinct \times perms of just the indistinct objects
Sort Semi-Distinct Objects

perms of distinct objects

perms of just the indistinct objects

= perms considering some objects are indistinct
Def: General Permutations:

When there are $n$ objects
$n_1$ are the same (indistinguishable) and
$n_2$ are the same and
...

$n_r$ are the same,

There are: $\frac{n!}{n_1!n_2!\ldots n_r!}$

Unique orderings ("permutations")
How many orderings?

\[
\text{Coke} \quad \text{Coke0} \quad \text{Coke} \quad \text{Coke0} \quad \text{Coke0}
\]

\[
= \frac{120}{(3! \times 2!)} = 10
\]
How many orderings of letters?

\[ \text{M00} = \frac{3!}{2!} = 3 \]

\[ \text{MISSISSIPPI} = \frac{11!}{1!4!4!2!} = 34,650 \]
Combinatorics

Counting tasks on $n$ objects

- Sort objects (permutations)
  - Distinct
    - $n!$
  - Some Distinct
    - $\frac{n!}{n_1!n_2!\ldots}$

- Choose $k$ objects (combinations)

- Put objects in $r$ buckets

Some Distinct

$n!$
How many unique 6 digit passcodes are there?

\[ 10^6 = 1,000,000 \]
How many possible codes 6 smudges?

If a phone password uses each of six distinct numbers, how many unique six digit passcodes are there?

6! = 720
How many possible codes 5 smudges?

If a phone password uses each of five distinct numbers, how many unique six digit passcodes are there?

Five mutually exclusive cases:
2 was repeated
4 was repeated
5 was repeated
6 was repeated
8 was repeated

\[ = 5 \times \frac{6!}{2!} = 1,800 \]
Combinatorics

Counting tasks on $n$ objects

- Sort objects (permutations)
  - Distinct
    - $n!$
  - Some Distinct
    - $\frac{n!}{n_1!n_2!\ldots}$
- Choose $k$ objects (combinations)
  - Distinct
- Put objects in $r$ buckets
Combinatorics

There are \( n = 20 \) people

How many ways can we choose \( k = 5 \) people to get cake?
Consider this generative process
Step 1: Randomly order people

There are $n = 20$ people

How many ways can we choose $k = 5$ people to get cake?

step 1 ways = $n!$
Step 2: Draw a line at pos $k$

There are $n = 20$ people

How many ways can we choses $k = 5$ people to get cake?

$\text{step 2 ways} = 1$
Step 3: Allow Cake Group to Mingle

There are $n = 20$ people

How many ways can we chose $k = 5$ people to get cake?

$k!$ different permutations lead to the same mingle
Step 4: Allow nonCake Group to Mingle

There are $n = 20$ people
How many ways can we choose $k = 5$ people to get cake?

$(n - k)!$ different permutations lead to the same mingle
Step 4: Allow nonCake Group to Mingle

There are $n = 20$ people
How many ways can we chose $k = 5$ people to get cake?

$$\text{num ways} = \frac{n! \times 1 \times 1}{k!(n-k)!}$$

- Randomly order $n$ objects
- Designate the first $k$ as chosen
- Any ordering of chosen group is the same choice
- Any ordering of non-chosen group is the same choice
Step 4: Allow nonCake Group to Mingle

There are $n = 20$ people

How many ways can we choose $k = 5$ people to get cake?

\[
\text{num ways} = \binom{n}{k}
\]

* Also called binomial coefficients

\[
(x + y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k \cdot y^{n-k}
\]
Counting tasks on $n$ objects

- **Sort objects** (permutations)
  - Distinct
  - $n!$

- **Choose $k$ objects** (combinations)
  - Some Distinct
  - \[ \frac{n!}{n_1!n_2! \ldots} \]

- **Put objects in $r$ buckets**
  - Distinct
  - \[ \binom{n}{k} = \frac{n!}{k!(n-k)!} \]
8,000 villagers. How many distinct ways can you \textbf{chose} 2 to play a game?

\[
\begin{align*}
\quad & = \frac{8000!}{7998!2!} = 31,996,000
\end{align*}
\]
Counting tasks on \( n \) objects

- **Sort objects (permutations)**
  - Distinct
  - \( n! \)
  - Some Distinct
  - \( \frac{n!}{n_1!n_2!\ldots} \)

- **Choose \( k \) objects (combinations)**
  - Distinct
  - \( \binom{n}{k} = \frac{n!}{k!(n-k)!} \)

- **Put objects in \( r \) buckets**
  - Distinct
  - None
  - Distinct
  - None Distinct
Combinatorics

Counting tasks on $n$ objects

- **Sort objects (permutations)**
  - Distinct
  - Some Distinct
  - Some Distinct
  - $n!$
  - $\frac{n!}{n_1!n_2!\ldots}$

- **Choose $k$ objects (combinations)**
  - Distinct
  - $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

- **Put objects in $r$ buckets**
  - Distinct
  - None Distinct
  - $r^n$
  - $\frac{(n + r - 1)!}{n!(r - 1)!}$
Something is going on in the world of AI
Modern AI
or, How we learned to combine probability and programming
Brief History
Early Optimism 1950

1952

1955

Axioms \[ \vdash C \]

ATP System (theorem prover)

Yes (proof/answer)

No

Timeout
Early Optimism 1950

“Machines will be capable, within twenty years, of doing any work a man can do.”
– Herbert Simon, 1952
The world is too complex

Underwhelming Results 1950s to 1980s

The spirit is willing but the flesh is weak.
(Russian)

The vodka is good but the meat is rotten.
BRACE YOURSELVES

WINTER IS COMING
Almost perfect...
What is going on?
[suspense]
Focus on one problem
Computer Vision

Piech, CS106A, Stanford University
Piech, CS106A, Stanford University

Classification

That is a picture of a one
Classification

Logistic Regression is like the Harry Potter Sorting Hat

That is a picture of a zero
Classification

That is a picture of an zero

* It doesn’t have to be correct all of the time
Can you do it?
What number is this?
What number is this?

1
How about now?

What a computer sees

```
0 0 1 0 1 0 1 0 0 0 1 1 1 0 1
1 0 0 1 0 1 1 1 0 1 0 0 0 1 0
1 1 1 0 1 0 0 1 1 0 0 1 0 1 0
1 1 1 1 1 0 0 0 0 0 1 1 0 1 1
0 0 0 1 1 0 0 1 0 0
1 0 0 1 1 0 0 0 1 0
1 1 0 1 1 0 0 1 1 0
1 0 1 0 0 1 0 0 1 0
0 0 0 0 1 0 1 0 1 1
0 1 1 0 0 0 0 0 1 1
0 0 1 0 1 1 1 0 0 0
0 1 1 1 0 1 0 0 1 0
1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1
0 0 1 1 1 0 1 0 1 1
```

What a human sees

Piech, CS106A, Stanford University
public class HarryHat extends ConsoleProgram {

    public void run() {
        println("Todo: Write program");
    }
}

Piech, CS106A, Stanford University
Two Great Ideas

1. Probability from Examples
2. Artificial Neurons
Two Great Ideas

1. Probability from Examples

2. Artificial Neurons
1. Probability From Examples
When Does the Magic Happen?

Lots of Data + Sound Probability
Machine Learning

Basically just a rebranding of statistics and probability.
Vision is Hard

You see this:

But the camera sees this:

[Andrew Ng]
Human Designed Features

To find edges at four orientations:

- Sum up edge strength in each quadrant.

Final feature vector:

- Human Features:
  - Find edges at four orientations
  - Sum up edge strength in each quadrant

[Andrew Ng]
Some Great Thinkers

Daphne Koller
Straight ML Not Perfect...
Two Great Ideas

1. Probability from Examples
2. Artificial Neurons
Two Great Ideas

1. Probability from Examples

2. Artificial Neurons
2. Artificial Neurons
Neuron

- Dendrites
- Soma
- Axon
- Myelin sheath
- Terminal button
Neuron

Diagram showing the structure of a neuron, including dendrites, soma, axon, myelin sheath, and terminal button.
Neuron
Some Inputs are More Important
Artificial Neuron

Piech, CS106A, Stanford University
Each node represents a neuron

Each edge represents the weight of the interaction
Neural Network

Each node represents a neuron

Each edge represents the weight of the interaction

Piech, CS106A, Stanford University
Each node represents a neuron.

Each edge represents the weight of the interaction.

Neural Network
Each node represents a neuron

Each edge represents the weight of the interaction

Neural Network
Interpret the last neuron as the “probability” that the image is of a 1.
The image had a 0 but we predicted a high probability that it was a 1.
Let’s update our weights to make our probabilities better match reality.

The image had a 0 but we predicted a high probability that it was a 1.
Let’s update our weights to make our probabilities better match reality.

The image had a 0 but we predicted a high probability that it was a 1.

Update the parameter at each connection.
Gradient of output layer params

\[
\frac{\partial L}{\partial \theta_i^{\hat{y}}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \theta_i^{\hat{y}}}
\]

\[
\hat{y} = \sigma \left( \sum_{j=0}^{m_h} h_j \theta_j^{\hat{y}} \right)
\]

\[
\frac{\partial \hat{y}}{\partial \theta_i^{\hat{y}}} = \sigma \left( \sum_{j=0}^{m_h} h_j \theta_j^{\hat{y}} \right) \left[ 1 - \sigma \left( \sum_{j=0}^{m_h} h_j \theta_j^{\hat{y}} \right) \right] \cdot \frac{\partial}{\partial \theta_i^{\hat{y}}} \sum_{j=0}^{m_h} h_j \theta_j^{\hat{y}}
\]

\[
= \hat{y}[1 - \hat{y}] \cdot \frac{\partial}{\partial \theta_i^{\hat{y}}} \sum_{j=0}^{m_h} h_j \theta_j^{\hat{y}}
\]

\[
= \hat{y}[1 - \hat{y}] \cdot h_i
\]

You will be able to do this.
Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012
A Neuron That Fires When It Sees Cats

Top stimuli from the test set

Optimal stimulus by numerical optimization

Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012
... smoothhound, smoothhound shark, Mustelus mustelus American smooth dogfish, Mustelus canis Florida smoothhound, Mustelus norrisi whitetip shark, reef whitetip shark, Triaenodon obesus Atlantic spiny dogfish, Squalus acanthias Pacific spiny dogfish, Squalus suckleyi hammerhead, hammerhead shark smooth hammerhead, Sphyrna zygaena smalleye hammerhead, Sphyrna tudes shovelhead, bonnethead, bonnet shark, Sphyrna tiburo angel shark, angelfish, Squatina squatina, monkfish electric ray, crampfish, numbfish, torpedo smalltooth sawfish, Pristis pectinatus guitarfish

rougtail stingray, Dasyatis centroura butterfly ray
eagle ray
spotted eagle ray, spotted ray, Aetobatus narinari cownose ray, cow-nosed ray, Rhinoptera bonasus manta, manta ray, devilfish

Atlantic manta, Manta birostris devil ray, Mobula hypostoma
grey skate, gray skate, Raja batis little skate, Raja erinacea...
ImageNet Classification

<table>
<thead>
<tr>
<th>0.005%</th>
<th>1.5%</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random guess</td>
<td>Pre Neural Networks</td>
<td>GoogLeNet</td>
</tr>
</tbody>
</table>

Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012
ImageNet Classification

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random guess</td>
<td>0.005%</td>
</tr>
<tr>
<td>Pre Neural Networks</td>
<td>1.5%</td>
</tr>
<tr>
<td>GoogLeNet</td>
<td>43.9%</td>
</tr>
</tbody>
</table>

Szegedy et al, Going Deeper With Convolutions, CVPR 2015
ImageNet Classification

0.005\%  1.5\%  82.7\%

Random guess  Pre Neural Networks  NASNet

Where is this useful?

A machine learning algorithm performs **better than** the best dermatologists.

Developed this year, at Stanford.

Open Problem: One Shot Learning

Human-level concept learning through probabilistic program induction.

Current deep learning methods are not enough to move the needle as far as we want, especially on socially relevant problems that often do not have the benefit of massive public datasets. The best new ideas are coming from probability theory.
Prediction: The person who solves one shot learning problem will use core probability
Closest thing to magic you can learn.
Now is the time, Stanford is the place.