The Large N Limit of the Knapsack Problem

From Bellman to Dantzig Through Boltzmann

Mobolaji Williams — *AE Data Science* — October 14 2024

Motivation and Outline

THE NOBEL PRIZE IN PHYSICS 2024

POPULAR SCIENCE BACKGROUND

8 October 2024

They used physics to find patterns in information

This year's laureates used tools from physics to construct methods that helped lay the foundation for today's powerful machine learning. ***John Hopfield*** *created a structure that can store and reconstruct information.* ***Geoffrey Hinton*** *invented a method that can independently discover properties in data and which has become important for the large artificial neural networks now in use.*

Motivation

How can we use physics to better understand computer science problems?

Motivation (Refined)

How can we use **statistical physics** to better understand the **Knaspack Problem?**

Knapsack Problem

What is the Knapsack Problem?

Statistical Physics

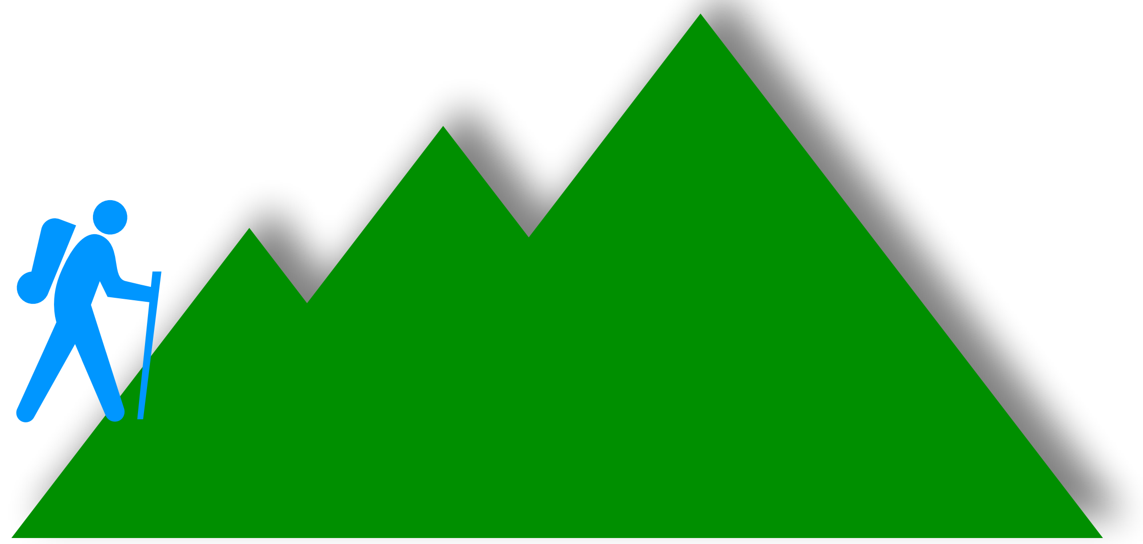
What is statistical physics?

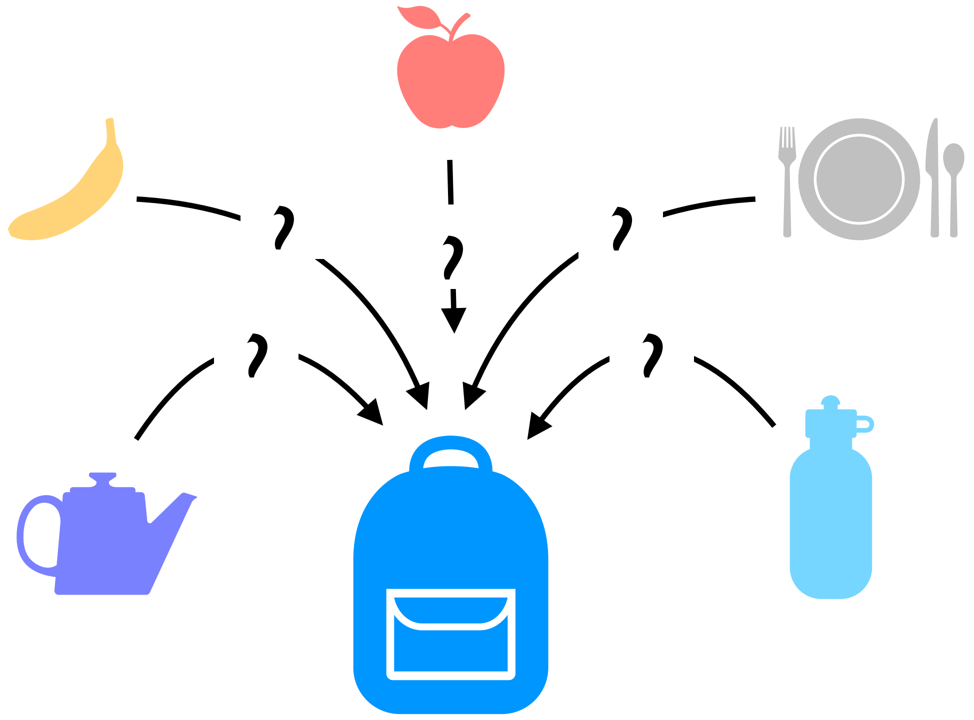
Statistical Physics & Knapsack Problem

How can statistical physics be applied to the Knapsack Problem?

What have we learned about the Knapack Problem?

Going on a Hike





- You're going on a hiking trip, and you need to pack a knapsack.
- The knapsack can hold a maximum weight of 20 kgs, and you need to fill it with valuable items.
- Say that you have a list of 10 items, each of which has a subjective value to you and a certain weight.

How can you fill the knapsack so that you **maximize the total value** of all the items inside while ensuring the **total weight doesn't exceed a given limit**?

This problem is known as the **Knapsack Problem.**

Object	Value	Weight
Apples	5	1 kg
Bananas	3	1 kg
Water	10	2 kg
Kettle	8	3 kg
Utensils	5	2 kg
⋮	⋮	⋮

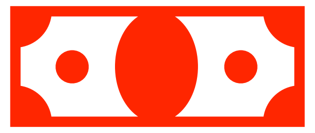
Weight Limit: 20 kg

* We ignore the "interaction value" between items where two items together are more valuable than each one separately (e.g., water and a kettle are more valuable)

This problem has various applications and manifestations

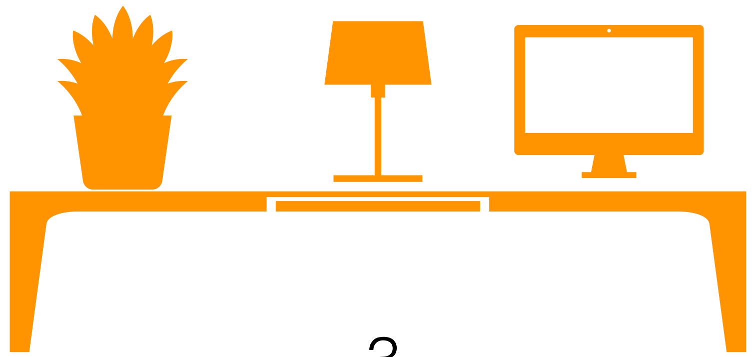
Investment Portfolio Optimization

How can we build a portfolio that maximizes total expected returns while total volatility remains below a given limit?



Shelf-Space Optimization

How can we choose items to place on a store shelf so as to maximize expected profit while not exceeding total shelf space?



Efficient Time Allocation

Which rides should we go on given our expected enjoyment, the amount of time it takes to ride, and the hours left in the day?

The Knapsack Problem: Abstraction

Knapsack Problem: How can you fill a knapsack so that you **maximize the total value** of all the items inside while ensuring the **total weight doesn't exceed a given limi**t?

Written more abstractly….

We have N objects labeled $j = 1,\ldots,N$.

Object j has weight w_j and value v_j.

$\mathbf{v} = (v_1, v_2, \ldots, v_N)$ vector of values

$\mathbf{w} = (w_1, w_2, \ldots, w_N)$ vector of weights

W weight limit

Problem Paramaters

$$X_j = \begin{cases} 1 & \text{if object } j \text{ is in solution} \\ 0 & \text{if object } j \text{ is not in solution} \end{cases}$$

$$\mathbf{X} = (X_1, X_2, \ldots, X_N)$$

Problem Solution

$\mathbf{X}\cdot\mathbf{v} = X_1v_1 + X_2v_2 + \cdots X_Nv_N$ (total value)

$\mathbf{X}\cdot\mathbf{w} = X_1w_1 + X_2w_2 + \cdots X_Nw_N$ (total weight)

How can we find $\mathbf{X}$?

What is the vector $\mathbf{X}$ that maximizes the quantity $\mathbf{X}\cdot\mathbf{v}$ subject to the constraint $\mathbf{X}\cdot\mathbf{w} \leq W$

Problem Statement

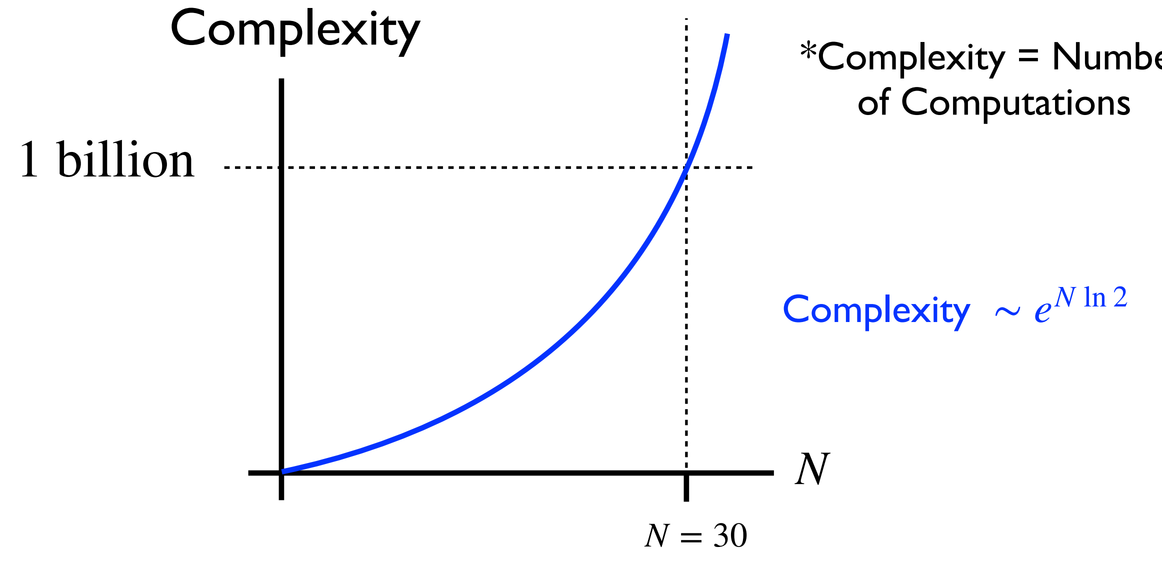
There are N objects and each one can either be inside or outside the knapsack…

… thus there are 2^N possible solutions, only some of which satisfy the weight limit.

Brute Force

(List all possible solutions and find the best)

1. List all possible 2^N solutions
2. Eliminate all the ones that violate the weight limit
3. Select the solutions with the maximum total value



*Complexity = Number of Computations

The number of computations increases **exponentially** with problem size.

Brute Force Approach is an inefficient method of solution

Object	Value	Weight
Apples	5	1 kg
Bananas	3	1 kg
Water	10	2 kg
Kettle	8	3 kg
Utensils	5	2 kg
⋮	⋮	⋮

Weight Limit: 20 kg

Written more abstractly…

Object	Value	Weight
1	v_1	w_1
2	v_2	w_2
3	v_3	w_3
4	v_4	w_4
5	v_5	w_5
⋮	⋮	⋮

Weight Limit: W

Dynamic Programming vs Greedy Solutions

What is the vector $\mathbf{X}$ that maximizes the quantity $\mathbf{X} \cdot \mathbf{v}$ subject to the constraint $\mathbf{X} \cdot \mathbf{w} \leq W$

Problem Statement

$\mathbf{v} = (v_1, v_2, \ldots, v_N)$ vector of values

$\mathbf{w} = (w_1, w_2, \ldots, w_N)$ vector of weights

W weight limit

Problem Paramaters

$$X_j = \begin{cases} 1 & \text{if object } j \text{ is in solution} \\ 0 & \text{if object } j \text{ is not in solution} \end{cases}$$

$$\mathbf{X} = (X_1, X_2, \ldots, X_N)$$

Problem Solution

How can we find $\mathbf{X}$?

Brute Force Solution

We can list all 2^N solutions, and select the one with the highest total value, subject to the weight limit.

The number of computations increases exponentially with problem size. → Brute Force Approach is an inefficient method of solution

There are two main alternative approaches from Computer Science 101

Dynamic Programming

(Recursively build from sub-solutions to the problem)

Richard Bellman (1953)

1. Define `V[N, W]` as the maximum value for the first `N` objects and a weight limit `W`

2. Note that

`V[N, W] = max{V[N-1, W],  v_N + V[N-1, W-w_N] }`

The Nth object is either included or not included in the solution

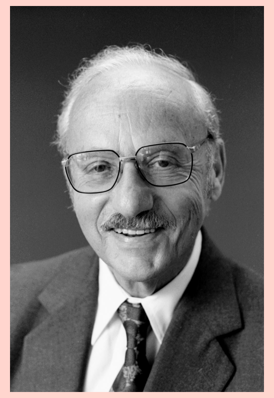
3. Using `V[0, 0] = 0` build up to the desired solution `V[N, W]`

(Exact Solution)

Complexity $\sim N \times W$

Greedy Solution

(Find all the items with the highest value to weight ratio)



George Dantzig (1948)

1. Compute v_j/w_j for all j

2. Sort the values in decreasing order of the ratio v_j/w_j

3. Fill up the knapsack with the highest ratio items, until the weight limit is reached

(Approximate Solution)

Complexity $\sim N \ln N$

Both of these approaches are taught in Computer Science classes as **distinct** ways of solving the Knapsack Problem

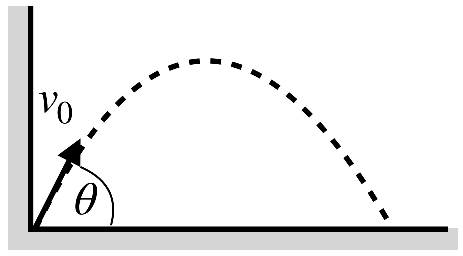
Introduction to Statistical Physics

Statistical Physics (def): Physics of systems with many degrees of freedom.

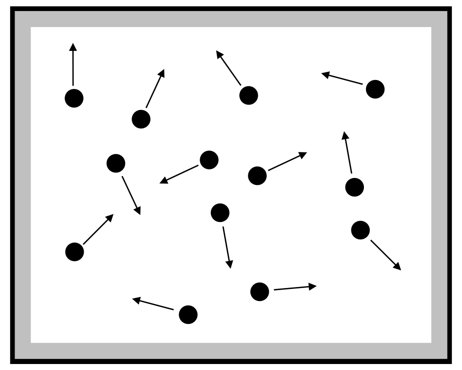
- In standard physics problems, you can track the dynamics of single particles
- But when you have on the order of 100 or more particles, you can't track the dynamics of particles individually…
- …so you have to use **statistical** properties like **averages** and **variances** to compute physical quantities

Projectile Motion

$$y(x) = (\tan\theta)x - \frac{gx^2}{2v_0^2\cos^2\theta}$$



N gas particles in a room



Partition Function: A way to compute probabilities for statistical physics systems

$$Z = \sum_{\text{states}} \exp\left[\underbrace{-E(\text{state})/T}_{\text{“Boltzmann Factor”}}\right]$$

Possible states of the system

Energy of system

System temperature

$$p(\text{state}) = \frac{1}{Z}\exp\left[-E(\text{state})/T\right] \quad \text{(State probability)}$$

Ludwig Boltzmann (1860)

* Connection to ML: The **softmax function** essentially is a **partition function** calculation

output layer	softmax activation	probabilities
$\begin{bmatrix}5\\7\\4\end{bmatrix}$ $\longrightarrow$	$\dfrac{e^{s_i/T}}{\sum_i e^{s_i/T}}$ $\longrightarrow$	$\begin{bmatrix}0.35\\0.45\\0.2\end{bmatrix}$

Example: Three state system

Particle can be in three states, each with a distinct energy

State:	1	2	3
Energy:	$-E_1$	$-E_2$	$-E_3$

$(E_1 > E_2 > E_3)$

What is the average energy of this system at $T = 0$?

$$Z = e^{E_1/T} + e^{E_2/T} + e^{E_3/T} \quad \text{(Partition Function)}$$

$$p_i = e^{E_i/T}/Z \quad \text{(State Probability)}$$

(Average Energy)

$$\langle E \rangle_T = \frac{E_1 e^{E_1/T} + E_2 e^{E_2/T} + E_3 e^{E_3/T}}{e^{E_1/T} + e^{E_2/T} + e^{E_3/T}} = \frac{E_1 + E_2 e^{(E_2-E_1)/T} + E_3 e^{(E_3-E_1)/T}}{1 + e^{(E_2-E_1)/T} + e^{(E_3-E_1)/T}}$$

$T \to 0$ $\quad (E_1 > E_2 > E_3)$

$= E_1$ $\quad$ (max value of E_i)

State Maxima from Statistical Averages:

By taking the average to $T \to 0$, we obtain the maximum value of the energy across states

$$\lim_{T\to 0} \langle E \rangle_T = E(\text{state})_{\text{max}}$$

Statistical Physics and the Knapsack Problem

Partition Function: A way to compute probabilities for statistical physics systems

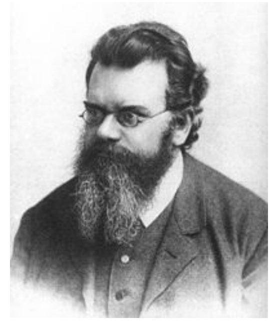
$$Z = \sum_{\text{states}} \overbrace{\exp\left[-E(\text{state})/T\right]}^{\text{"Boltzmann Factor"}}$$

Possible states of the system

Energy of system

System temperature

$$p(\text{state}) = \frac{1}{Z}\exp\left[-E(\text{state})/T\right] \quad \text{(State probability)}$$



Ludwig Boltzmann (1860)

State Maxima from Statistical Averages:

By taking the average to $T \to 0$, we obtain the maximum value of the energy

$$\lim_{T\to 0}\langle E\rangle_T = E(\text{state})_{\text{max}}$$

Main Question: How can we use statistical physics to solve the Knapsack Problem?

Define the "states" as the various solutions to the knapsack problem, and define the "energy" in terms of the total value of these states

$$X_j = \begin{cases} 1 & \text{if object } j \text{ is in solution} \\ 0 & \text{if object } j \text{ is not in solution} \end{cases}$$

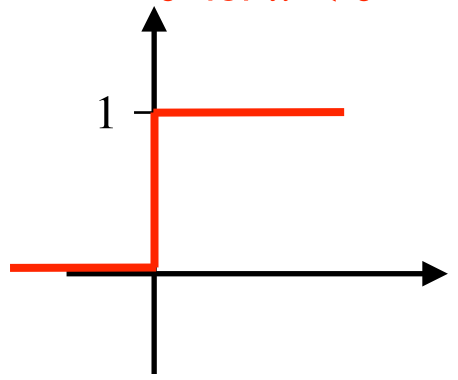
$$\mathbf{X} = (X_1, X_2, \ldots, X_N)$$

$$\text{Energy} = -\mathbf{X}\cdot\mathbf{v} = -\left(X_1 v_1 + \cdots + X_N v_N\right)$$

Only include states that satisfy the weight limit $W \geq \mathbf{X}\cdot\mathbf{w}$

Step Function

$$\Theta(x) = \begin{cases} 1 & \text{for } x \geq 0 \\ 0 & \text{for } x < 0 \end{cases}$$



$\beta \equiv 1/T$ (Inverse temperature)

$$\underbrace{Z_N(\beta\mathbf{v}, \mathbf{w}, W)}_{\text{Knapsack Problem Partition Function}} = \underbrace{\sum_{X_1=0}^{1}\cdots\sum_{X_N=0}^{1}}_{\text{Summation over all states}} \underbrace{\Theta\left(W - \mathbf{X}\cdot\mathbf{w}\right)}_{\text{Step Function}} \underbrace{\exp(\beta\,\mathbf{X}\cdot\mathbf{v})}_{\text{Boltzmann Factor}}$$

We want to use this partition function to compute…

… the average value $\mathbf{X}\cdot\mathbf{v}$ across all states…

…then the maximum value across all states.

$$\langle\mathbf{X}\cdot\mathbf{v}\rangle = \frac{1}{Z_N(\beta\mathbf{v}, \mathbf{w}, W)}\frac{\partial}{\partial\beta}Z_N(\beta\mathbf{v}, \mathbf{w}, W)$$

$$V_N(W) \equiv \lim_{\beta\to\infty}\langle\mathbf{X}\cdot\mathbf{v}\rangle$$

*Since $\beta = 1/T$, taking $T \to 0$ is the same as taking $\beta \to \infty$

To do this we need to introduce two mathematical techniques

I. Complex Analysis

II. Gaussian Approximations

Statistical Physics leads to Dynamic Programming

Main Question: How can we use statistical physics to solve the Knapsack Problem?

Need to introduce two mathematical techniques

I. Complex Analysis*

II. Integral Approximations

*Calculus on the complex plane (e.g., $z = x + iy$ where $i = \sqrt{-1}$)

$$Z_N(\mathbf{v}/T, \mathbf{w}, W) = \sum_{X_1=0}^{1} \cdots \sum_{X_N=0}^{1} \Theta\,(W - \mathbf{X}\cdot\mathbf{w})\exp(\mathbf{X}\cdot\mathbf{v}/T)$$

Knapsack Problem Partition Function

Summation over all states

Step Function

Boltzmann Factor

Complex Analysis: Step function identity

$$\Theta(k-\ell) = \frac{1}{2\pi i}\oint_{\Gamma} \frac{dz}{z^{k+1}}\frac{z^{\ell}}{1-z} = \begin{cases} 1 & \text{for } k \geq \ell \\ 0 & \text{for } k < \ell \end{cases}$$

Average value across all states

$$\langle \mathbf{X}\cdot\mathbf{v}\rangle = \frac{1}{Z_N(\beta\mathbf{v}, \mathbf{w}, W)}\frac{\partial}{\partial\beta}Z_N(\beta\mathbf{v}, \mathbf{w}, W) \quad \#1)$$

…then the maximum value across all states.

$$V_N(W) \equiv \lim_{\beta\to\infty}\langle \mathbf{X}\cdot\mathbf{v}\rangle \quad \#2)$$

$$\boxed{Z_N(\beta\mathbf{v}, \mathbf{w}, W) = \frac{1}{2\pi i}\oint\frac{dz}{z^{W+1}}\frac{1}{1-z}\prod_{j=1}^{N}\left(1 + z^{w_j}e^{\beta v_j}\right)}$$

$\prod_{j=1}^{N} a_j = a_1 \times a_2 \times \cdots \times a_N$

(Product notation)

Isolate $j = N$ factor

$$= \frac{1}{2\pi i}\oint\frac{dz}{z^{W+1}}\frac{1}{1-z}\left(1 + z^{w_N}e^{\beta v_N}\right)\prod_{j=1}^{N-1}\left(1 + z^{w_j}e^{\beta v_j}\right)$$

Distribute terms

$$\boxed{Z_N(\beta\mathbf{v}, \mathbf{w}, W) = Z_{N-1}\left(\beta\mathbf{v}, \mathbf{w}, W\right) + e^{\beta v_N}Z_{N-1}\left(\beta\mathbf{v}, \mathbf{w}, W - w_N\right)} \quad \#3)$$

Recursive Partition Function Identity

Reminiscent of the "Dynamic Programming" solution to the Knapsack Problem

V[N, W] = max{V[N-1, W], v_N + V[N-1, W-w_N] }

Not a coincidence!

Combining

#1) #2) #3)

Yields

$$V_N(W) = \max\left\{V_{N-1}(W), v_N + V_{N-1}(W - w_N)\right\}$$

Dynamic Programming Solution

Main Point: The **Statistical Physics** representation of the Knapsack Problem leads to the **Dynamic Programming Solution**.

Ludwig Boltzmann

Statistical Physics

Recursive Identity

Richard Bellman

Dynamic Programming

Statistical Physics leads to Greedy Solutions

Main Question: How can we use statistical physics to solve the Knapsack Problem?

Need to introduce two mathematical techniques

I. Complex Analysis

II. Gaussian Approximations*

*Ways to take complex integrals and simplify them

Average value across all states

$$\langle \mathbf{X}\cdot\mathbf{v}\rangle = \frac{1}{Z_N(\beta\mathbf{v},\mathbf{w},W)}\frac{\partial}{\partial\beta}Z_N(\beta\mathbf{v},\mathbf{w},W) \quad \#1)$$

…then the maximum value across all states.

$$V_N(W) \equiv \lim_{\beta\to\infty}\langle \mathbf{X}\cdot\mathbf{v}\rangle \quad \#2)$$

$$\Theta(x) = \begin{cases} 1 & \text{for } x \geq 0 \\ 0 & \text{for } x < 0 \end{cases}$$

$$Z_N(\beta\mathbf{v},\mathbf{w},W) = \sum_{X_1=0}^{1}\cdots\sum_{X_N=0}^{1}\Theta\left(W-\mathbf{X}\cdot\mathbf{w}\right)\exp(\beta\,\mathbf{X}\cdot\mathbf{v})$$

$$= \frac{1}{2\pi i}\oint\frac{dz}{z^{W+1}}\frac{1}{1-z}\prod_{j=1}^{N}\left(1+z^{w_j}e^{\beta v_j}\right)$$

$$F_N(z;\mathbf{v}/T,\mathbf{w},W) \equiv -W\ln z - \ln(1-z) + \sum_{j=1}^{N}\ln\left(1+zw^je^{v_j/T}\right)$$

$$= \frac{1}{2\pi i}\oint\frac{dz}{z}\exp F_N(z;\beta\mathbf{v},\mathbf{w},W)$$

Large N Gaussian Approximation $N \gg 1$

$$Z_N(\beta\mathbf{v},\mathbf{w},W) \simeq \frac{\exp F_N(z_0;\mathbf{v}/T,\mathbf{w},W)}{\sqrt{2\pi\, z_0^2\,\partial_z^2 F_N(z;\mathbf{v}/T,\mathbf{w},W)}} \quad \#3)$$

Approximate Partition Function

Combining

#1) #2) #3)

Maximum Value *

$$V_N(W) = \sum_{j=1}^{N} v_j\,\Theta(v_j/w_j-\gamma_0)$$

where γ_0 is defined as

$$W = \sum_{j=1}^{N} w_j\,\Theta(v_j/w_j-\gamma_0)$$

Only include an object if it's **value-to-weight** ratio is greater than γ_0

$$X_j = \Theta(v_j/w_j-\gamma_0)$$

γ_0 is the **minimum value-to-weight ratio** of objects included in the knapsack

Filling up the knapsack based on the value-to-weight ratio up to a given ratio…

…is the essence of the **Greedy Solution** to the Knapsack Problem

Recall

Greedy Solution

(Find all the items with the highest value to weight ratio)

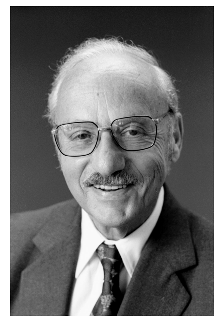
1. Compute v_j/w_j for all j
2. Sort the values in decreasing order of the ratio v_j/w_j
3. Fill up the knapsack with the highest ratio items, until the weight limit is reached

Main Point: The Large N approx. of the **Statistical Physics** representation of the Knapsack Problem leads to the **Greedy Solution**.

Ludwig Boltzmann

Statistical Physics

$\xrightarrow[\text{approximation}]{\text{Large } N}$



George Dantzig

Greedy Solution

From Bellman to Dantzig through Boltzmann

Main Question: How can we use statistical physics to solve the Knapsack Problem?

<u>The Knapsack Problem</u>

What is the vector $\mathbf{X}$ that maximizes the quantity $\mathbf{X} \cdot \mathbf{v}$ subject to the constraint $\mathbf{X} \cdot \mathbf{w} \le W$

We demonstrated this for the 0-1 Knapsack Problem

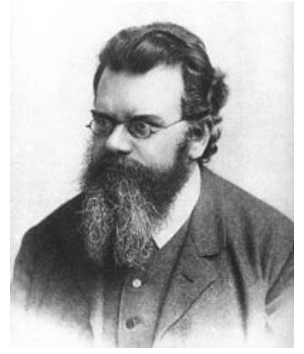
But we can demonstrate it for other **Combinatorial Optimization*** problems

(Def)* Finding an optimal object across a discrete set of objects

- Number Partitioning Problem
- Task Assignment Problem
- Coin Change Problem

⋮

This **generality** is important because the Knapsack Problem is part of a larger **computational complexity class** of problems



Ludwig Boltzmann (1860)

Statistical Physics

$$Z_N(\beta \mathbf{v}, \mathbf{w}, W) = \frac{1}{2\pi i} \oint \frac{dz}{z^{W+1}} \frac{1}{1-z} \prod_{j=1}^{N} \left(1 + z^{w_j} e^{\beta v_j}\right)$$

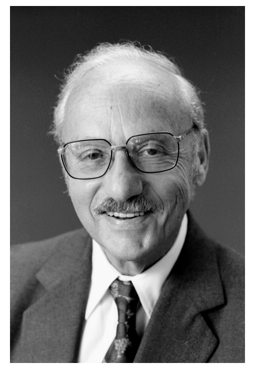
Recursive Identity

Large N approximation

Richard Bellman (1953)

Dynamic Programming

$$V_N(W) = \max\left\{V_{N-1}(W), v_N + V_{N-1}(W - w_N)\right\}$$



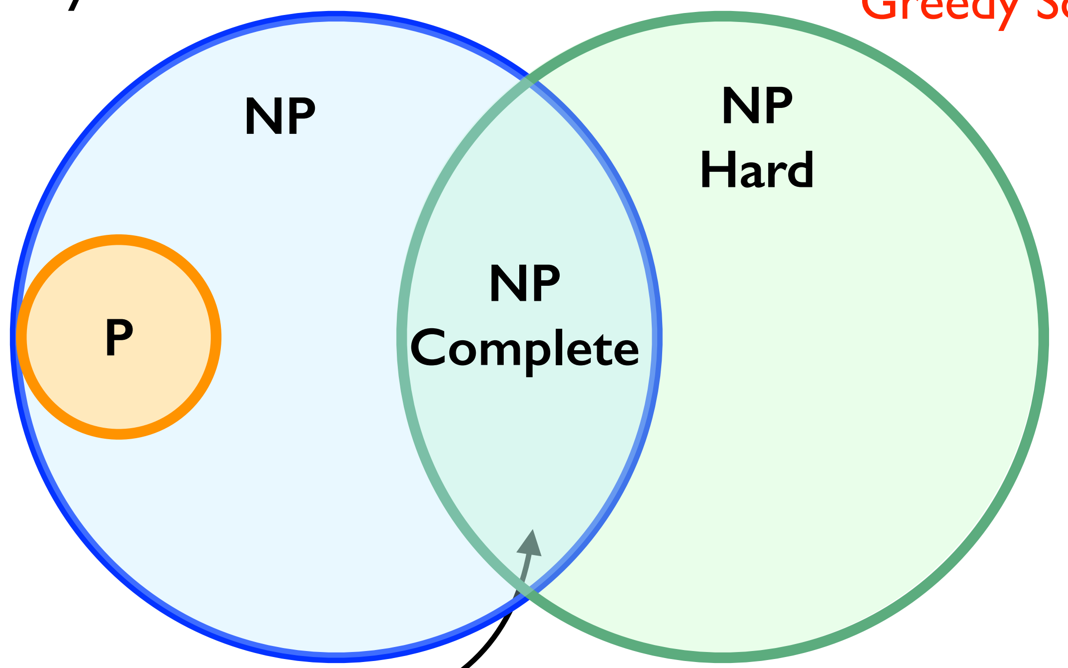
George Dantzig (1948)

Greedy Solution

$$X_j = \Theta(v_j/w_j - \gamma_0)$$

γ_0 is the minimum value-to-weight ratio of objects included in the knapsack

Computational Complexity Classes



Knapsack Problem

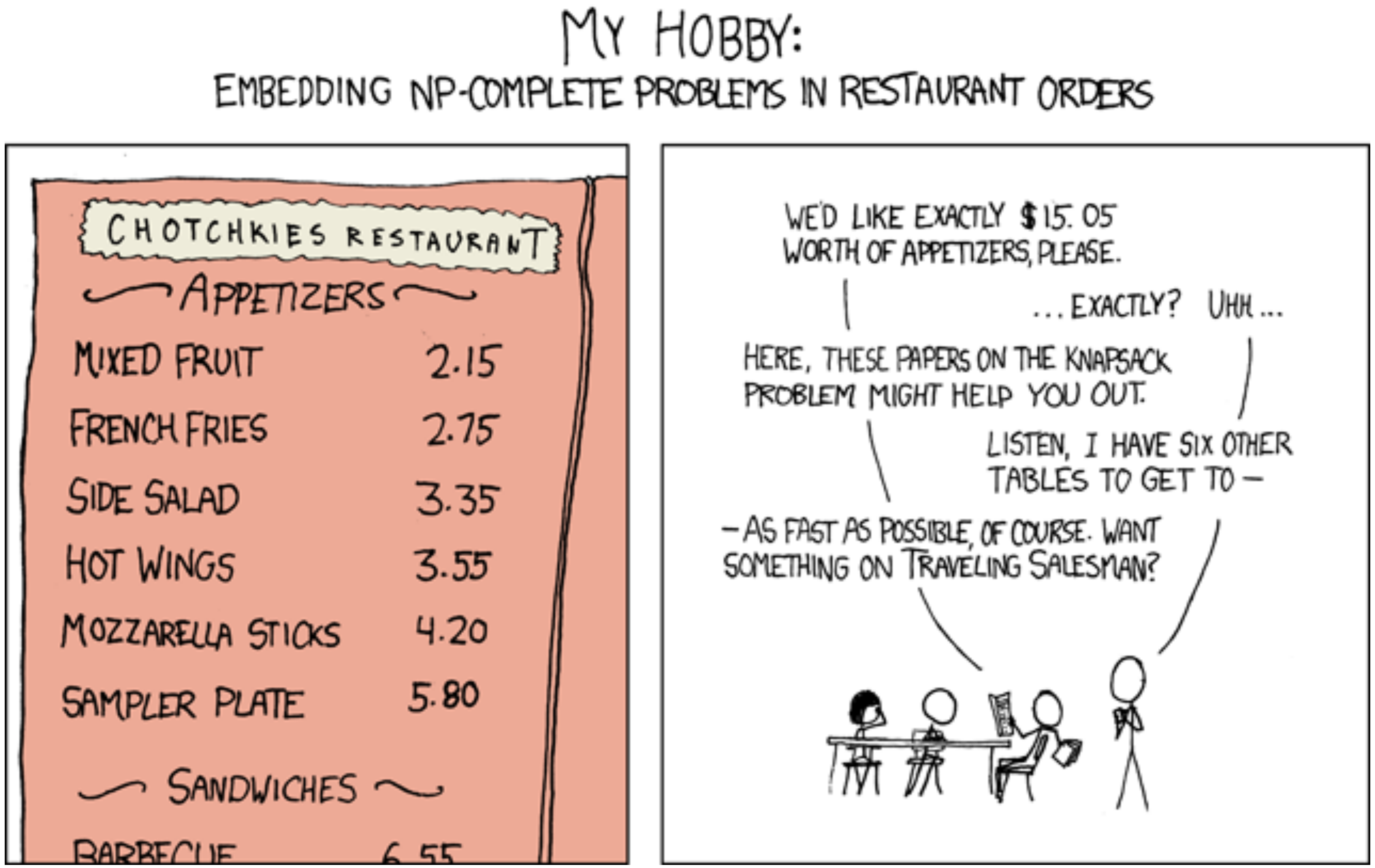
Motivation (Refined)

How can we use **statistical physics** to better understand the **Knaspack Problem?**

Conclusion: We can connect the **Dynamic Programming** and **Greedy Solution** to the Knapsack Problem by formulating the problem as a **Statistical Physics** system

Concluding Question: Can we take what we have learned about the Knapsack Problem (and other Comb Opt problems) and say something about the limits of Computational Complexity Classes?

Ending Slide Requisite XKCD comic



https://xkcd.com/287/

Resources

- History
    - **Richard Bellman** (https://mathshistory.st-andrews.ac.uk/Biographies/Bellman/)
    - **George Dantzig** (https://mathshistory.st-andrews.ac.uk/Biographies/Dantzig_George/)
    - **Ludwig Boltzmann** (https://mathshistory.st-andrews.ac.uk/Biographies/Boltzmann/)
- Theory
    - **Knapsack Problem Applications** (https://en.wikipedia.org/wiki/Knapsack_problem)
    - **Meaning of Complexity Classes** (https://stackoverflow.com/questions/1857244/what-are-the-differences-between-np-np-complete-and-np-hard)
- Source Material
    - **Paper** :"Large W limit of the Knapsack Problem," MW PRE 2024: https://mowillia.github.io/documents/PhysRevE.109.044151.pdf
    - **Github Repo**: https://github.com/mowillia/LargeWKP