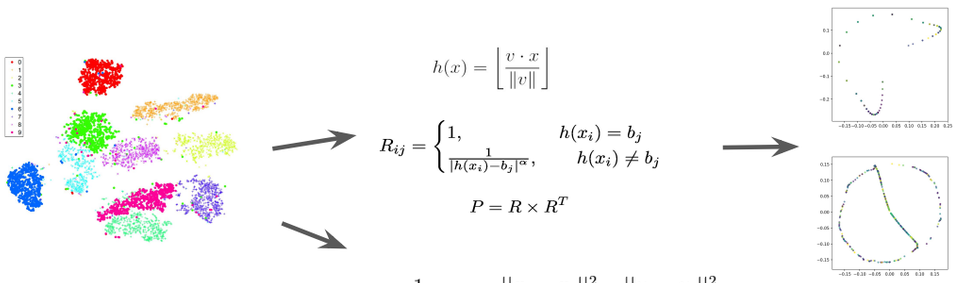
Can we make t-SNE faster using Locality Sensitive Hashing or designing a new cost function?



$$h(x) = \left\lfloor \frac{v \cdot x}{\|v\|} \right\rfloor$$

$$R_{ij} = \begin{cases} 1, & h(x_i) = b_j \\ \frac{1}{|h(x_i) - b_j|^{\alpha}}, & h(x_i) \neq b_j \end{cases}$$

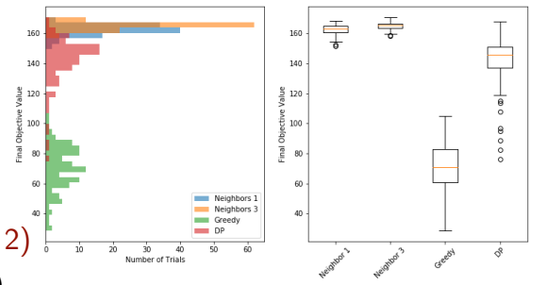
$$P = R \times R^T$$

$$cost = \frac{1}{2} \sum_i \sum_j \frac{||x_i - x_j||^2 - ||y_i - y_j||^2}{||x_i - x_j||^2}$$

Experiments with t-SNE

(Riley Hadjis, Suyog Soti, Aparajithan Venkateswaran)

```
procedure SIMULATED ANNEALING(ℙ(E_i, E_j, T), N(x), T(k), k_max)
    x ← x_0                                              ▷ Initialize x
    for k = 1, ..., k_max do
        T ← T(k)                                         ▷ Update temperature, based on k
        x_candidate ← N(x)                               ▷ Propose a new candidate state
        if ℙ(E(x), E(x_candidate), T) ≥ Uni(0,1) then    ▷ Possibility of state change
            x ← x_candidate
        end if
    end for
    return x
end procedure
```

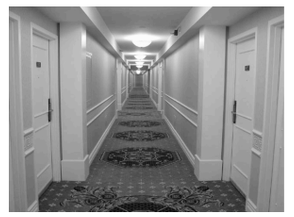


Simulated Lineup Smithing

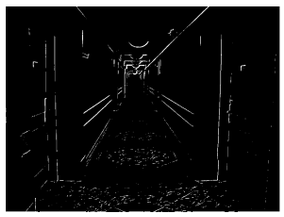
(Draft Kings portfolio selection, ver. 2)

(Tyler Schuessler, Nelson Mitchell)

Algorithm	Max Objective Value	Mean Objective Value	Mean Runtime (s)
Greedy	104.85	70.047	0.341
Dynamic Programming	167.98	141.48	82.4
Branch & Bound (n=1)	168.51	N/A	N/A
Neighbors 1 SA	170.87	162.38	0.46
Neighbors 3 SA	170.87	164.96	0.673

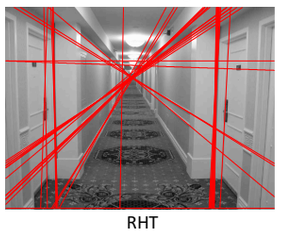


Origin Image



Edge Image

HT



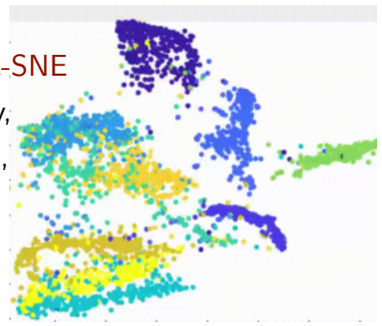
RHT

Hough Transformation and its Randomized Version: Line Detection

(Jiafan Qian)

Improvements to t-SNE

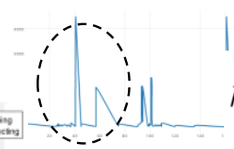
(Kathryn Gray, Ryan Marizza, Zack Jensen)



https://github.com/stephenbeckr/randomized-algorithm-class

APPM 4720/5720 (special topics) "Randomized Algorithms" Prof. Becker, spring 2019

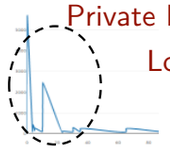
Student projects



$$h(x) = \left\lfloor \frac{a^\top \mathcal{T} x + b}{r} \right\rfloor$$

Private IoT Device Identification Using Locality-Sensitive Hashing

(Will Shand)



Student backgrounds:

- Applied Math (BS, BS/MS, PhD)
- Engineering Physics (BS)
- Computer Science (BS)
- Electrical & Comp. Eng. (BS)
- Aerospace (PhD)

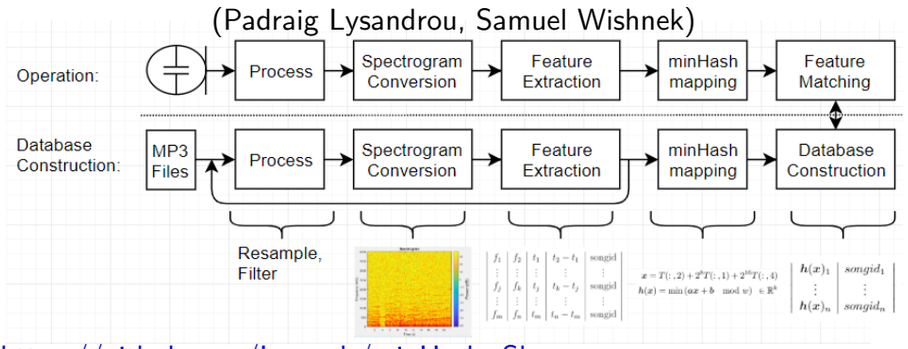
Algorithm	AKS	Miller-Rabin(k=5)	Fermat(k=5)
Time(s)	9.632	26.313	20.729
Accuracy	1000/1000	1000/1000	1000/1000

An Analysis of Primality Tests

(Adrian Strock)

Music Clip Identification with Randomized Locality Sensitive Hash Tables

(Padraig Lysandrou, Samuel Wishnek)



https://github.com/Lysandr/minHash_Shazam

Figure 1: Block diagram describing the numerical processes of our algorithm.

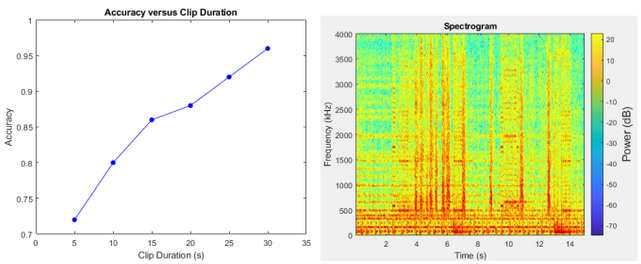


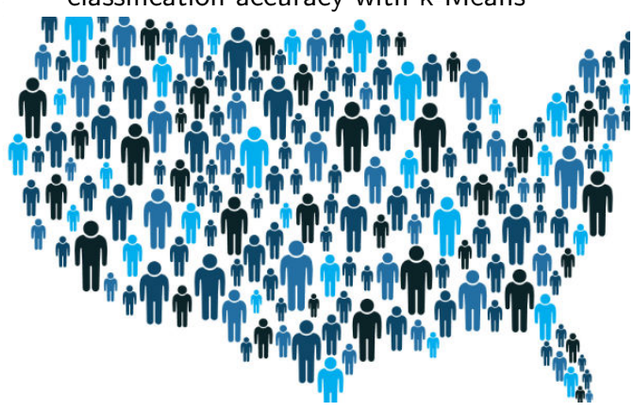
Figure 2: A clip of Darude - Sandstorm after processing and spectrogram conversion.

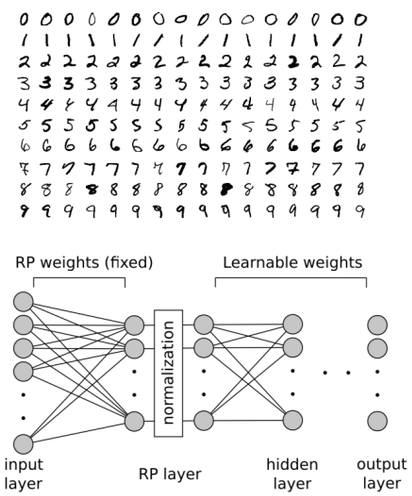
Classifying Income

(Cheryl Hansen)

Use census data to predict if annual income is above or below $50K

- Quantifying categorical data for use in k-Means++
- Subsampling entries while obtaining similar classification accuracy with k-Means





Training Neural Nets with Random Projections

(Alec Dunton, Fortino Garcia, Felix Newberry)

Projection	100 × 100	300 × 100	1000 × 1000
No Sketch	97.30	98.06	98.16
Gaussian	96.26 (.92)	96.36 (.88)	96.84 (.80)
Subsampling	92.37 (.90)	92.85 (.87)	92.67 (.82)
FJLT	97.97 (.92)	97.57 (.89)	97.50 (.84)
Sparse	96.60 (.92)	97.19 (.89)	97.41 (.79)
CountSketch	96.61 (.90)	97.23 (.87)	97.48 (.80)