Visualizing High Dimensional Data using Parallelized t-SNE Dimensionality Reduction

Hugo Rivera

Input and Output

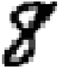
$$x = \{x_1, x_2, x_3, \ldots, x_N\}$$

$$x_i \in \mathbb{R}^H$$

$$y = \{y_1, y_2, y_3, \ldots, y_N\}$$

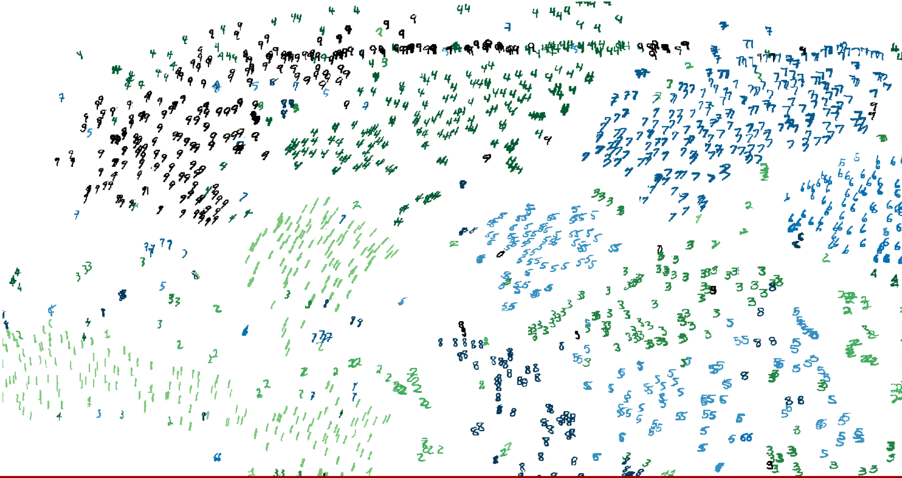
$$y_i \in \mathbb{R}^2$$

Example input $x_i \in \mathbb{R}^{784}$

$x_1 =$  , $x_2 =$  , $x_3 =$  , $\ldots$

Example for $y_i \in \mathbb{R}^2$

Example



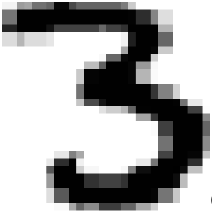
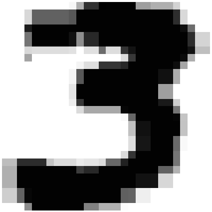
The t-SNE Algorithm

Student's **t**-Distributed

Stochastic

Neighbor Embedding

Neighbor Embedding

$x_i =$  close to $x_j =$ 

$\implies$

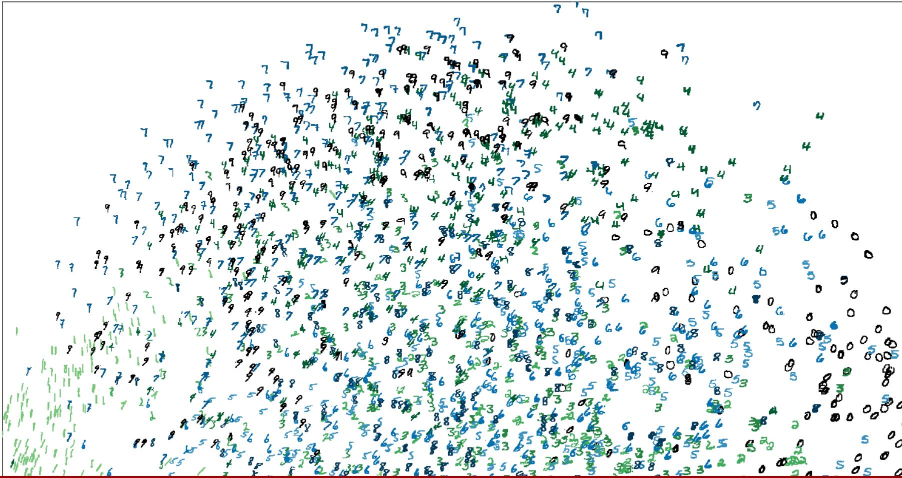
y_i close to y_j

Algorithm Pseudocode 1

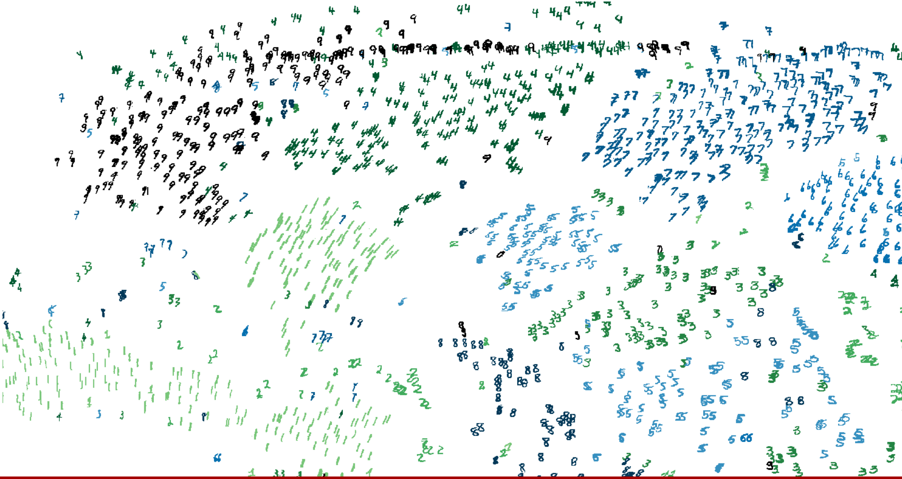
van der Maaten and Hinton 2008 [Journal of Machine Learning]

1. Guess y.
2. Iteratively improve y.

Demo



Demo



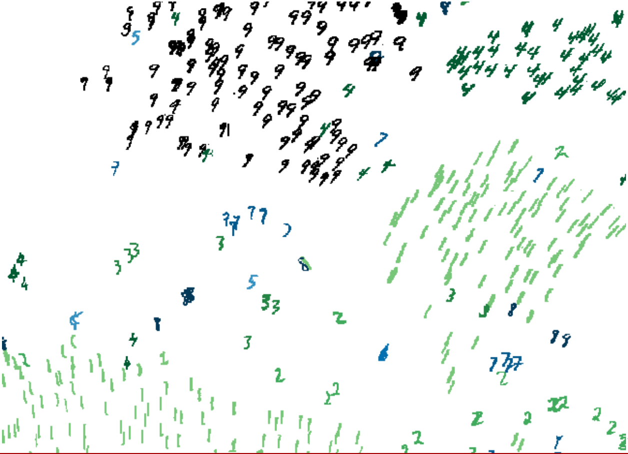
N-Body System

van der Maaten 2014 [Journal of Machine Learning]

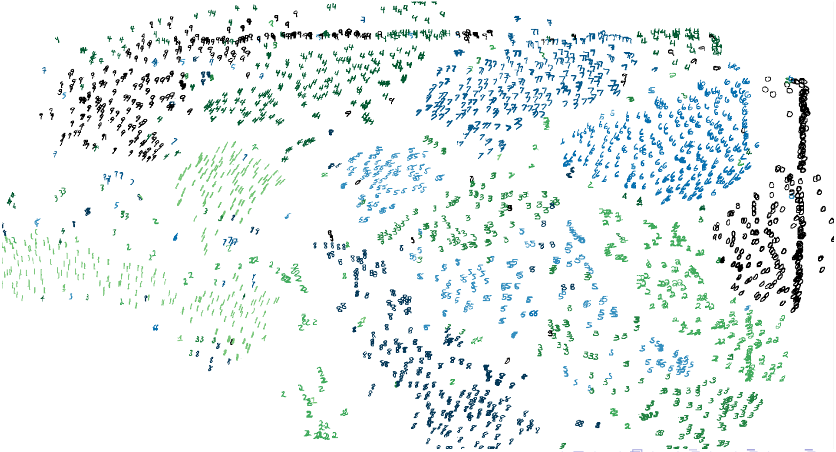
Connected by springs

Spring strength $\sim \text{dist}_1(x_i, x_j) - \text{dist}_2(y_i, y_j)$

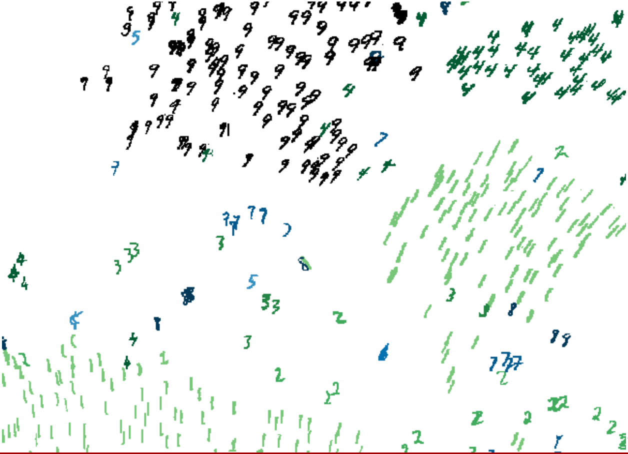
Demo



Demo

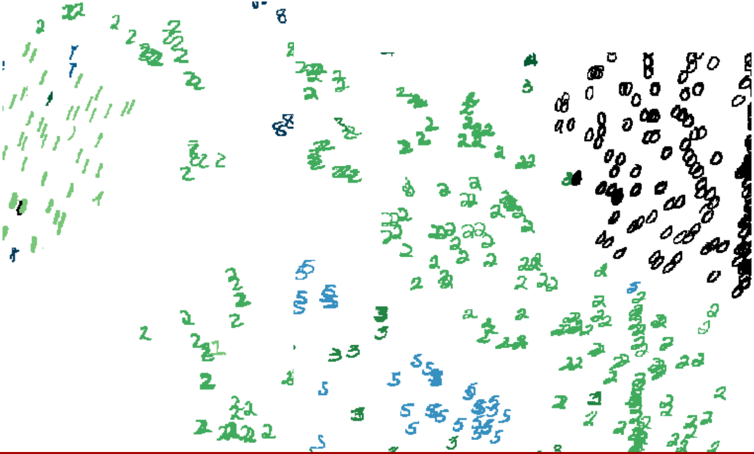


Demo



Demo

Demo



Demo

http://infohost.nmt.edu/~hrivera/hpc/mnist.mp4

My Code

```
hr@y580 > cloc tsne.c matrix.c
Language       comment    code
---------------------------
C + OpenMP     216        470
```

$O(n^2/p)$

Algorithm Pseudocode 2

```
P = N x N matrix
P[i,j] = probability that x[i] neighbors x[j]
y[i] = random 2D vector from -1E-4 to 1E-4
for t=1..T
 for i=1..N
  Qy = N x N matrix
  Qy[i,j] = probability that y[i] neighbors y[j]
  y[i] = y[i] - gradient of KL(P, Qy)
```

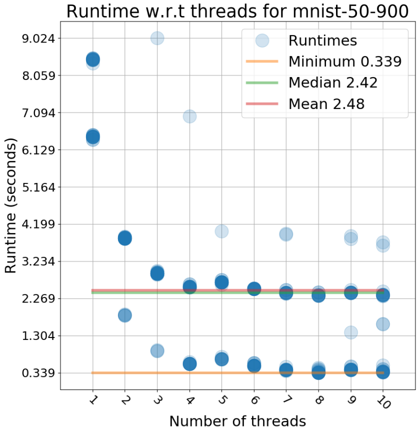
Parallelized using Hogwild GD

Niu, Recth, Re, Wright 2011

```
for t=1..T
 parallel for i=1..N shared(P, Qy, y)
  Qy = N x N matrix
  Qy[i,j] = probability that y[i] neighbors y[j]
  y[i] = y[i] - gradient of KL(P, Qy)
```

Best if sparse

Performance Increase



How? (part 2)

van der Maaten and Hinton 2008 [Journal of Machine Learning]

Compute the pairwise similarities of x, $P \in \mathbb{R}^{N \times N}$.

Guess y, compute pairwise similarities $Q(y) \in \mathbb{R}^{N \times N}$.

Must minimize difference between P and $Q(y)$.

Minimize Difference Using

Gradient Descent

Kullbach-Leibler divergence: diff of P, Q.

$$KL(P,Q) = \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} P_{ij} \log \frac{P_{ij}}{Q_{ij}}$$

$$y_i^{\text{improved}} = y_i - \frac{\delta}{\delta y_i} KL(P, Q(y))$$

Equations

$$P_{j|i}(x) = \frac{\exp\left(-\|x_i - x_j\|^2/2\sigma_i^2\right)}{\sum_{k=1,k\neq i}^{N}\exp\left(-\|x_i - x_k\|^2/2\sigma_i^2\right)}$$

$$P_{ij} = (P_{j|i} + P_{i|j})/2N$$

$$Q_{ij}(y) = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k=1}^{N}\sum_{l=1,l\neq k}^{N}(1 + \|y_k - y_l\|^2)^{-1}}$$

Equations

$$KL(P,Q) = \sum_{i=1,j\neq i}^{N} P_{ij} \log \frac{P_{ij}}{Q_{ij}}$$

$$\frac{\delta}{\delta y_i} KL(P,Q(y)) = \sum_{j\neq i}^{N} 4(P_{ij} - Q_{ij}$$

$$)(y_i - y_j) Q_{ij} Z$$

Barnes-Hut Approximation

Group distant nodes

Questions?