Introduction to t-SNE

Stats 302 Research Presentation

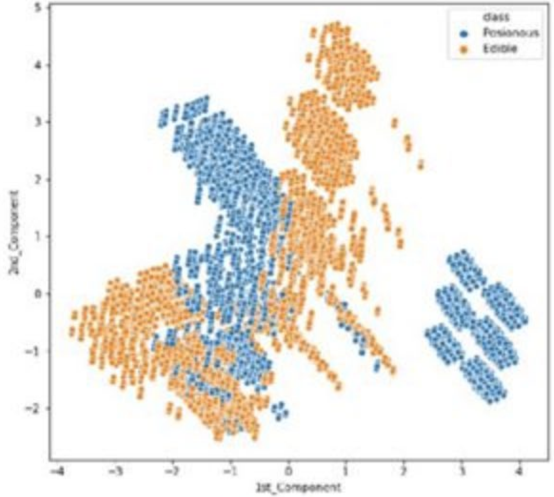
Lexue Song, Xi Chen, Rongfan Liu

Imagine you are a biologist.........

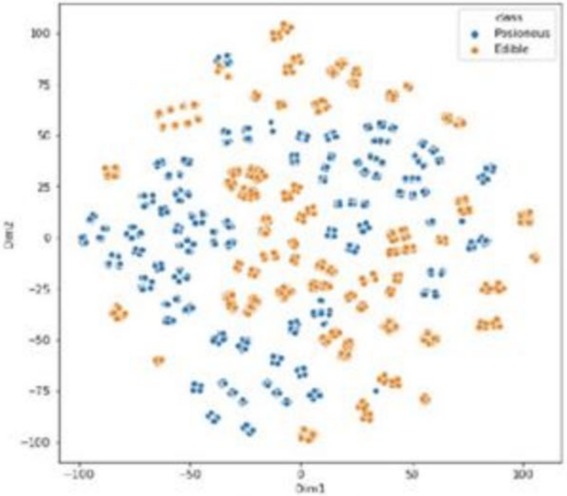
Introduction of t-SNE

- t-Distributed Stochastic Neighbor Embedding
- an unsupervised, non-linear technique
- visualize high-dimensional data in a two-dimensional space

t-SNE vs PCA



PCA

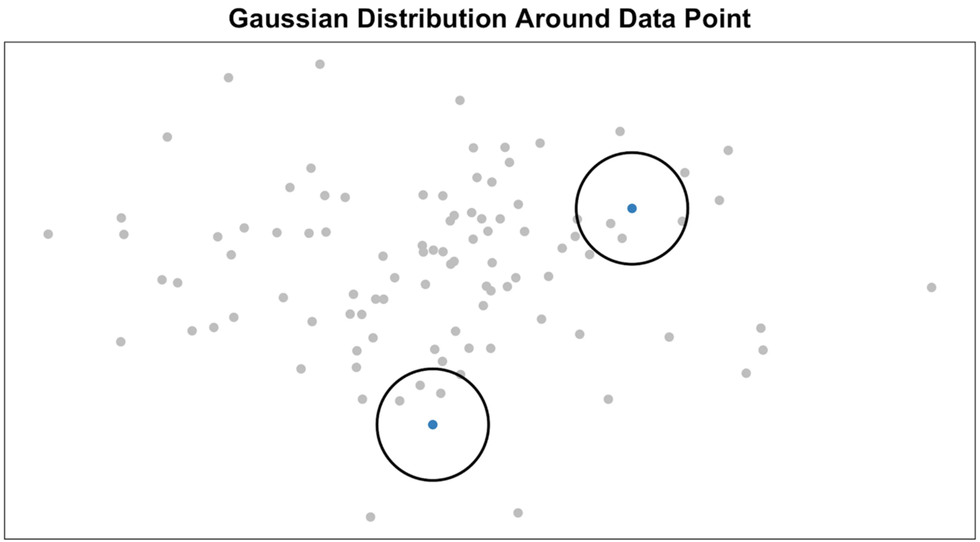


t-SNE

The t-SNE algorithm clusters the poisonous and edible mushrooms without any overlap.

Algorithm of T-SNE： Pairwise similarity

1.Compute pairwise similarities between data points in the high-dimensional space using Gaussian probability distribution



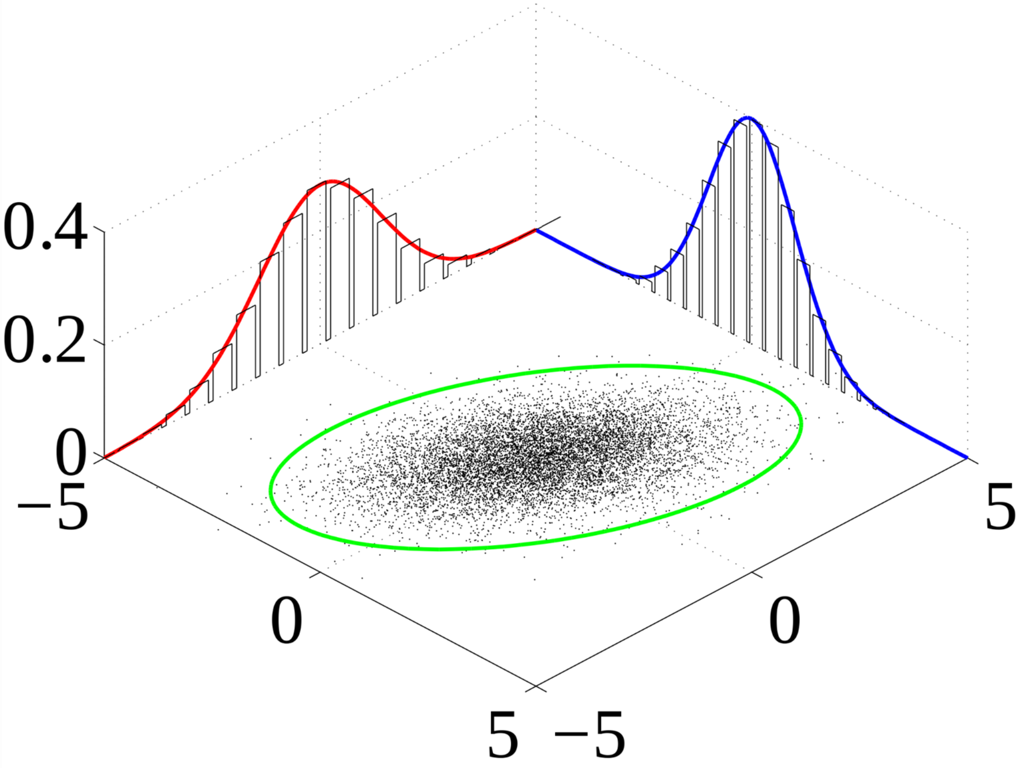
Algorithm of T-SNE： Similarity computation

We first calculate the Euclidean distance between the two points.

Then we then use this distance to calculate the pairwise similarity between the two points using a Gaussian distribution:

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}$$

Algorithm of T-SNE： Similarity computation

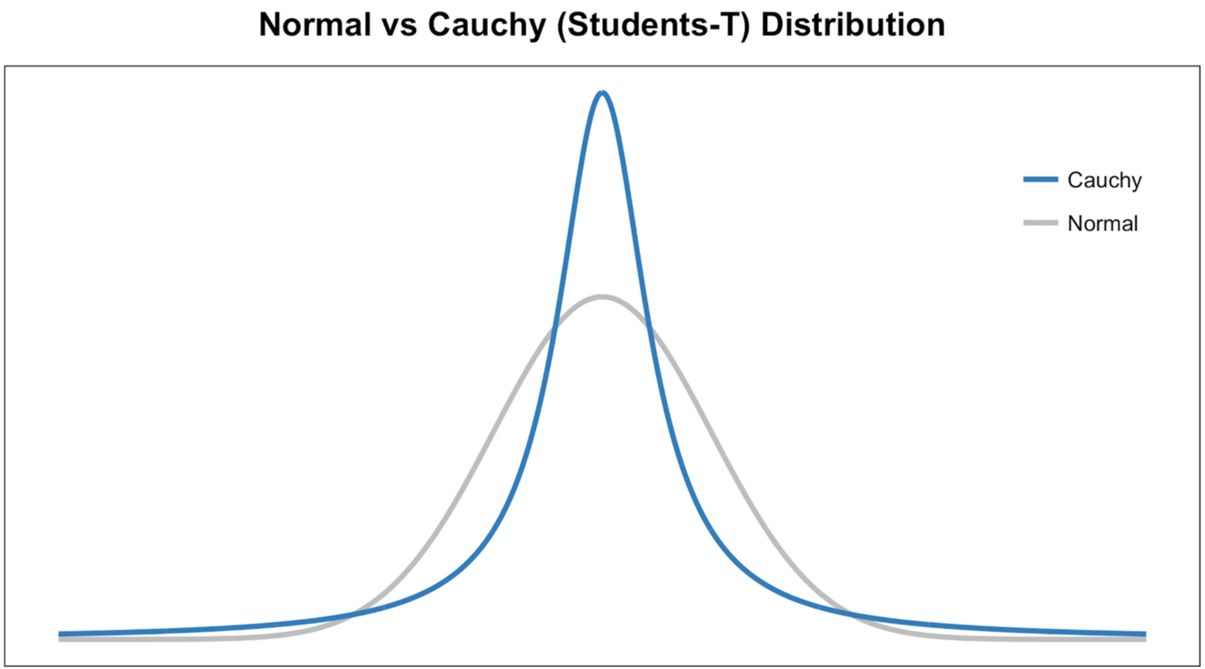


We can repeat this process for all pairs of data points. These pairwise similarities are then converted into a joint probability distribution that represents the similarities between all data points.

Algorithm of T-SNE

Initialize the data points in a lower-dimensional space: The algorithm constructs a lower-dimensional space, typically 2 or 3 dimensions, and randomly initializes the data points in this space.

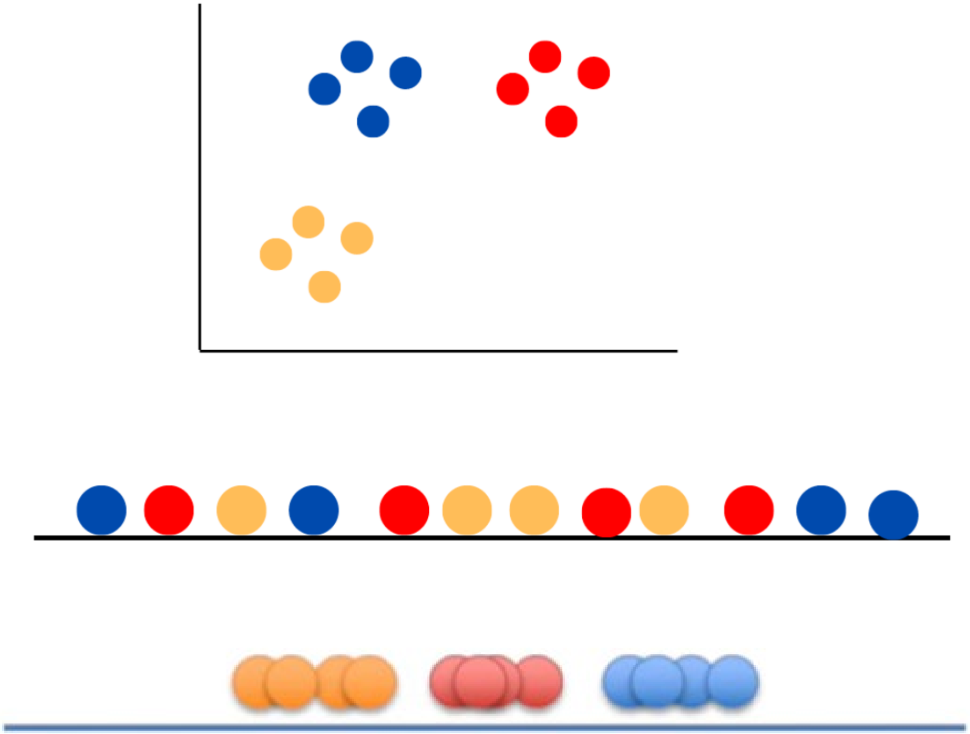
Algorithm of T-SNE： Similarity computation



Instead of Gaussian distribution, we use student-t distribution in low dimension.

Heavier tails can bear outliners.

Algorithm of T-SNE： Dimensionality Reduction



Algorithm of T-SNE： Kullback–Leibler Divergence

$$D_{\mathrm{KL}}(P\|Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}.$$

for discrete probability distributions

$$D_{\mathrm{KL}}(P\|Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x,$$

for continous probability distributions

Algorithm of T-SNE： Gradient descent

The gradient of the loss function is :

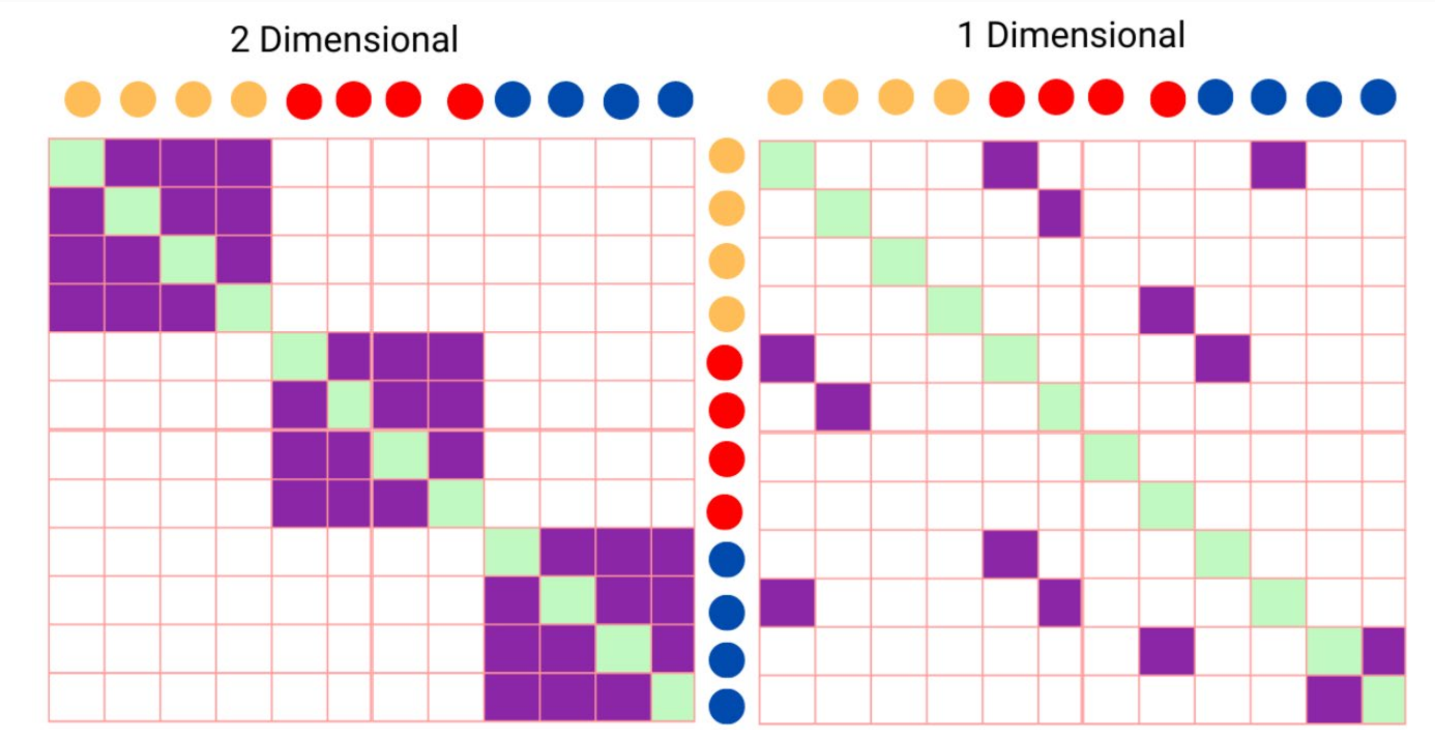
$$\frac{\partial C}{\partial y_i} = 2 \sum_{j \neq i} (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(y_i - y_j)$$

Gradient Update in each iteration :

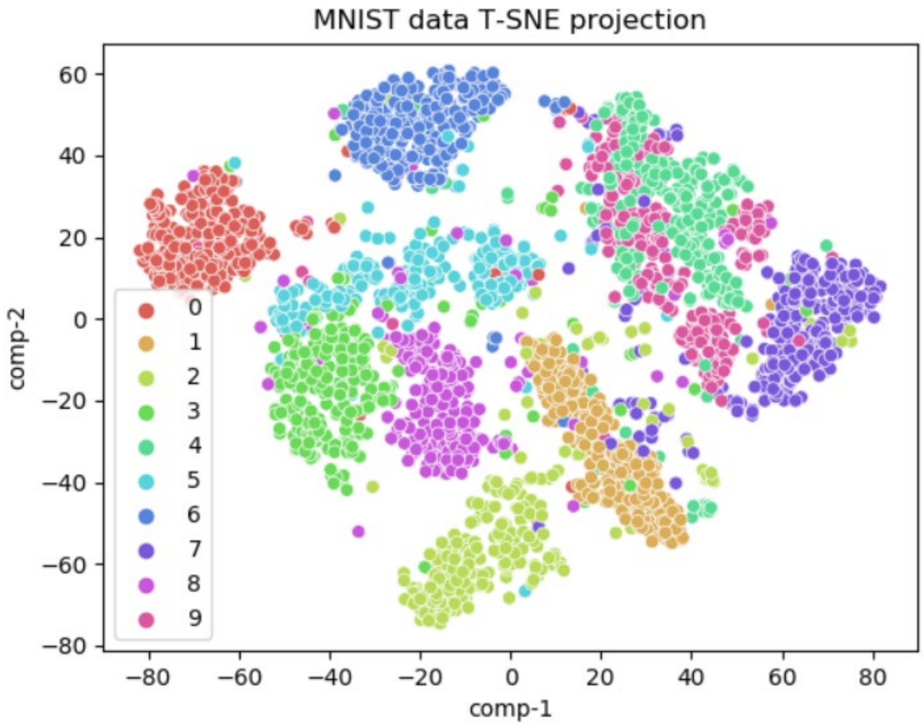
$$\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta y} + \alpha(t)\left(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)}\right),$$

where $\alpha(t)$: *Momentum at iteration t*
$\mathcal{Y}^{(t)}$: *Solution at iteration t*
η : *Learning Rate*

Algorithm of T-SNE： Similarity matrix



T-SNE: application and demo



References

https://newmortalbeing.medium.com/understanding-t-sne-d869ece441a5

https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a

https://www.datatechnotes.com/2020/11/tsne-visualization-example-in-python.html

https://en.wikipedia.org/wiki/Joint_probability_distribution#/media/File:Multivariate_normal_sample.svg

https://www.zhihu.com/search?type=content&q=t-SNE%E5%92%8CPCA

Thank you!