If your PCA looks weird
and it don't look good,
Who you gonna call?
t-SNE!

A Brief Introduction to
Dimensionality Reduction with t-SNE

Don Perkus

4/26/17

PCA – Dimensionality Reduction

- How it works
  - Projecting on to best hyper-plane
  - Minimize distances of far separated points
- Limitations
  - Linear algorithm, so it can't interpret complex non-linear relationships between features.
  - Focus on placing dissimilar data points far apart in a lower dimension representation.

t-SNE

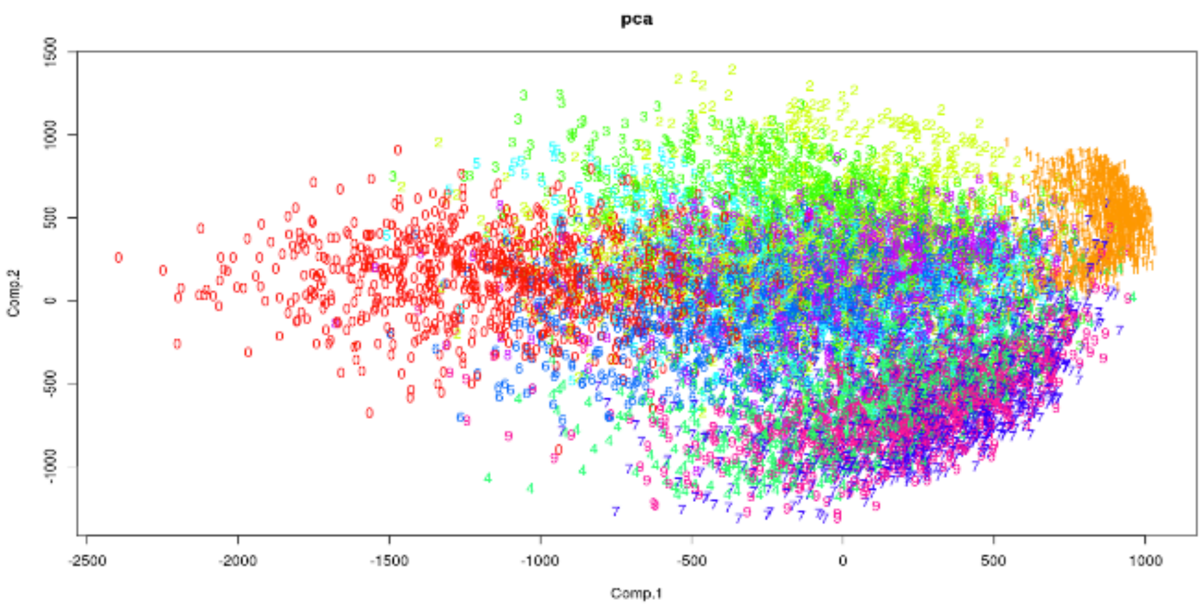
- t-Distributed Stochastic Neighbor Embedding
- Often it is important that similar data points be represented close together.
- t-SNE is based on probability distributions with gradient descent on neighborhood graphs to find the structure within the data.

Example: handwritten digits dataset

From https://www.slideshare.net/ssuserb667a8/visualization-data-using-tsne

PCA Digits Visualization

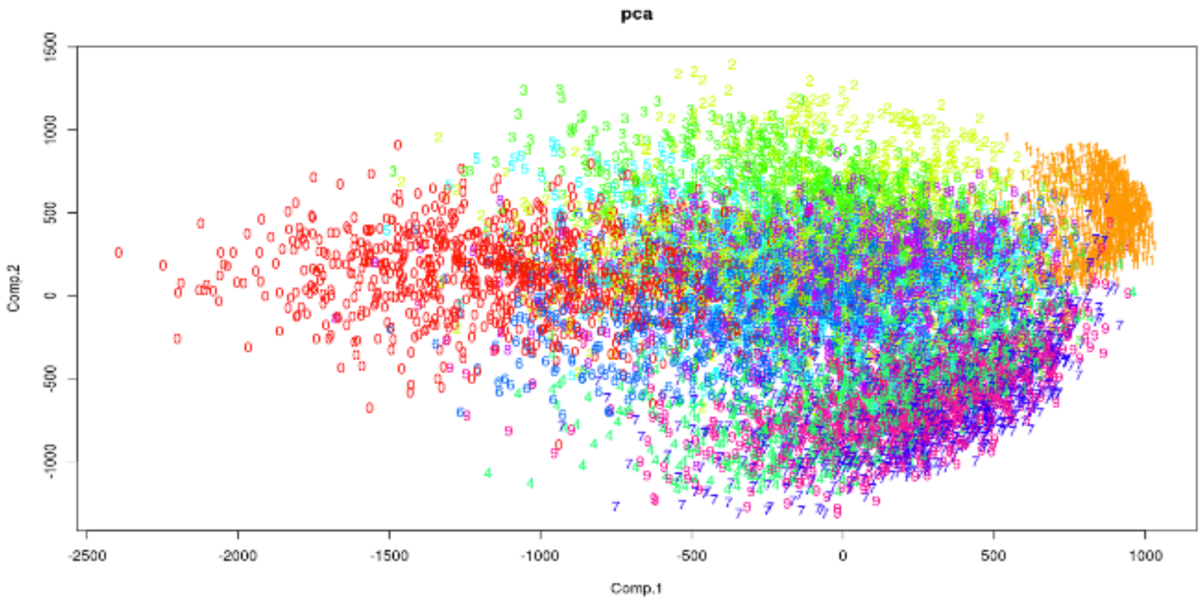
PCA



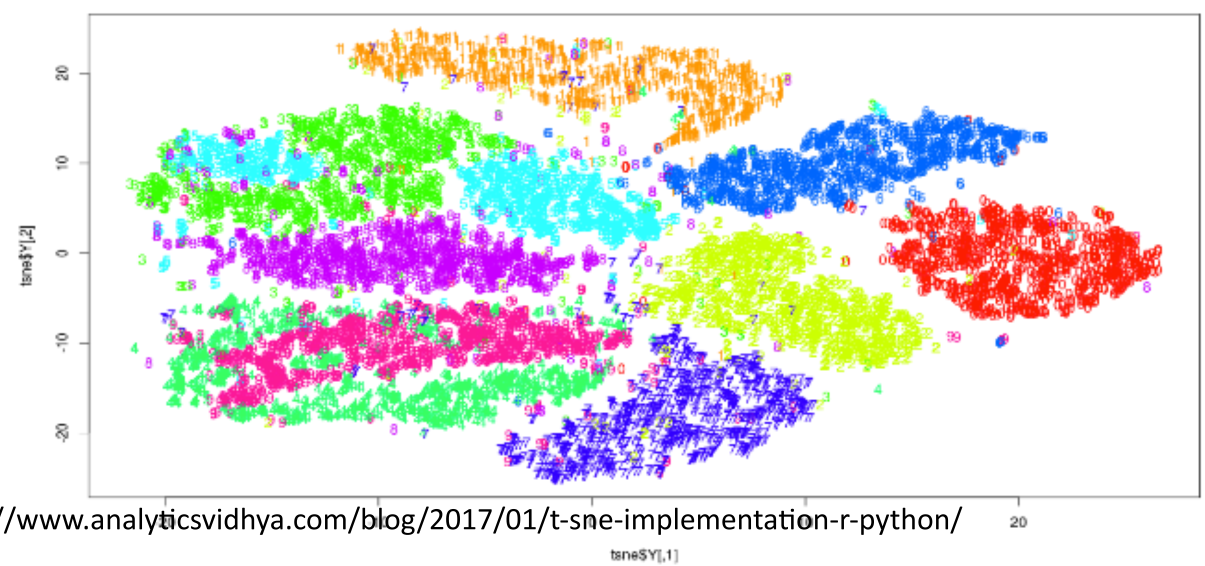
From https://www.analyticsvidhya.com/blog/2017/01/t-sne-implementation-r-python/

PCA vs t-SNE Digits Visualization

PCA
11 sec



t-SNE
118 sec

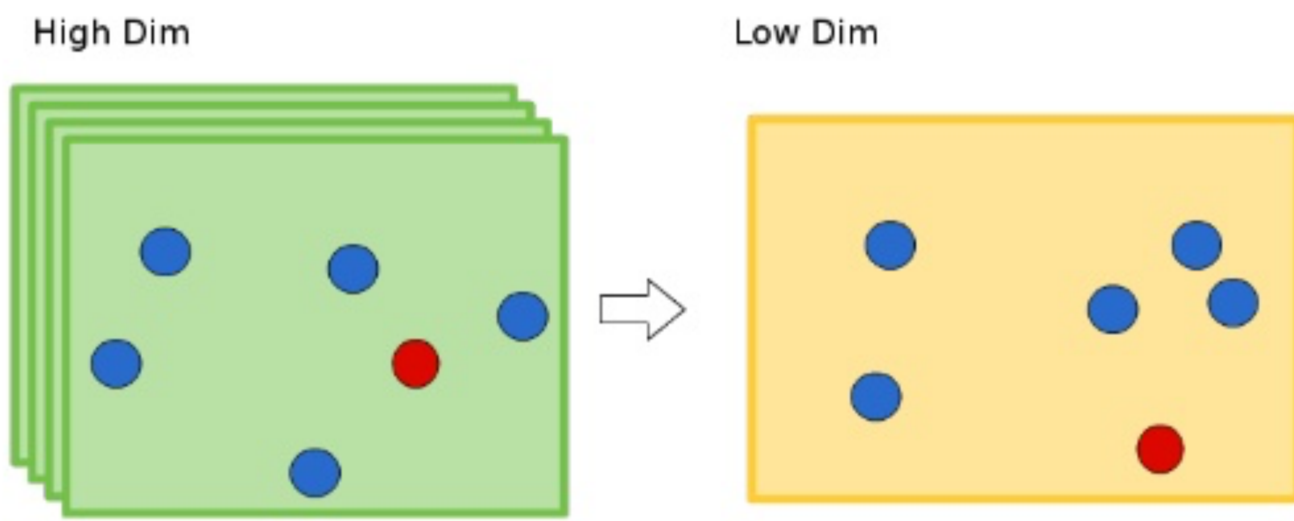


From https://www.analyticsvidhya.com/blog/2017/01/t-sne-implementation-r-python/

t-SNE Algorithm

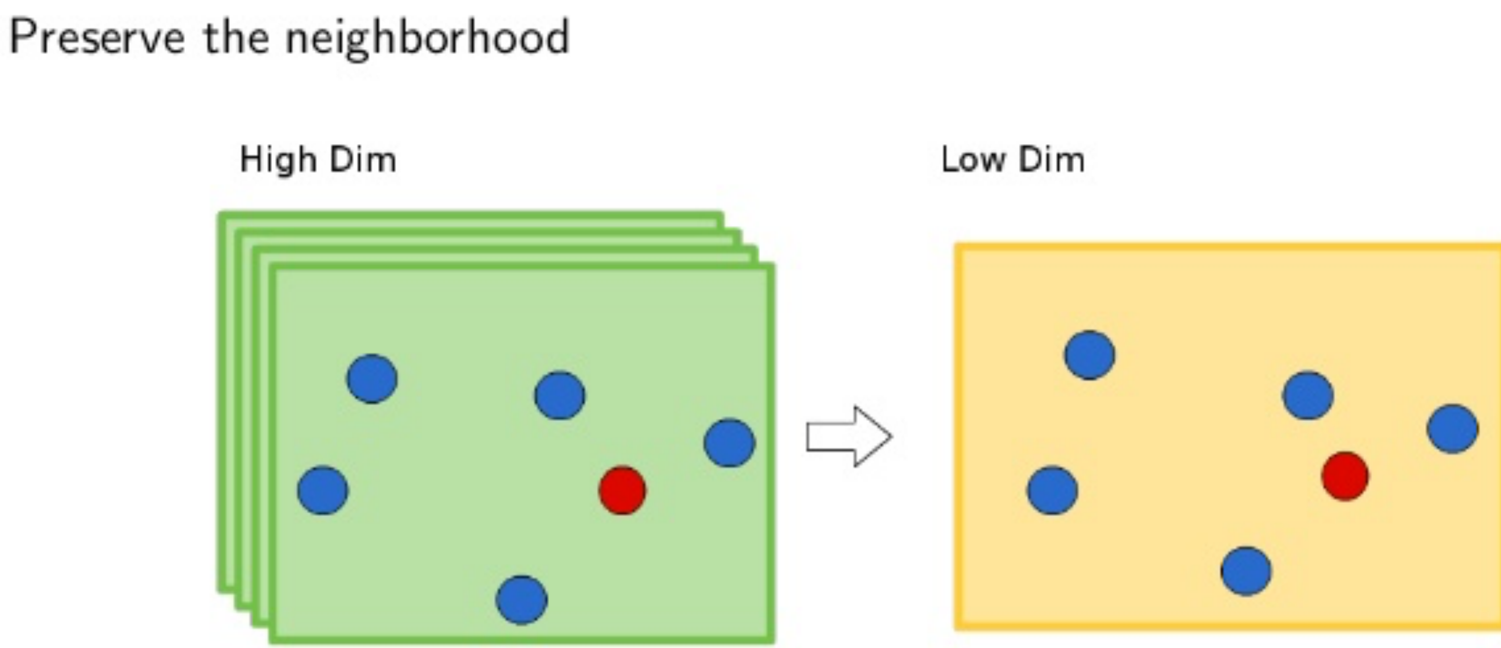
- Calculate the conditional probability of similarity between each pair of points in
  - High dimensional space
  - Low dimensional space
- Compute a cost function
  - Close neighbors should stay close
  - Middle and far neighbors may vary
  - “Normalized” by local density
- Gradient Descent

Map points – initially not so good



From https://www.slideshare.net/ssuserb667a8/visualization-data-using-tsne

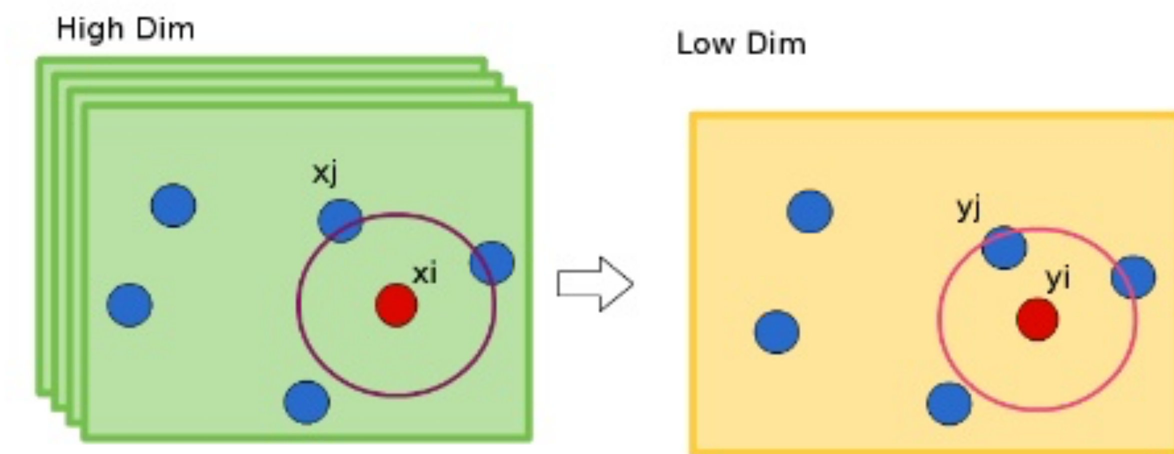
Map points - better



From https://www.slideshare.net/ssuserb667a8/visualization-data-using-tsne

Measure Pairwise Similarities
Create a Similarity Matrix

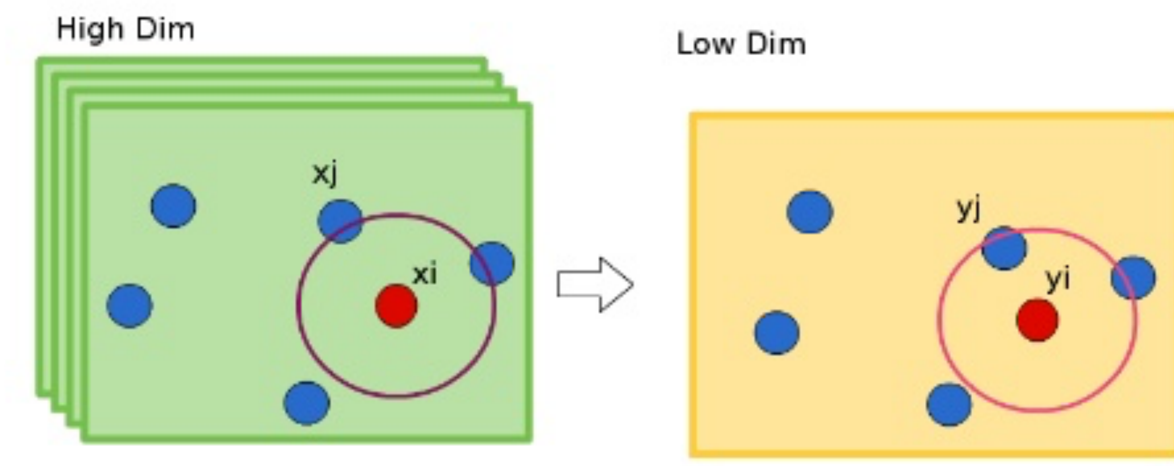
Measure pairwise similarities between high-dimensional and low-dimensonal objects



From https://www.slideshare.net/ssuserb667a8/visualization-data-using-tsne

Create a Similarity Matrix Conditional Probabilities

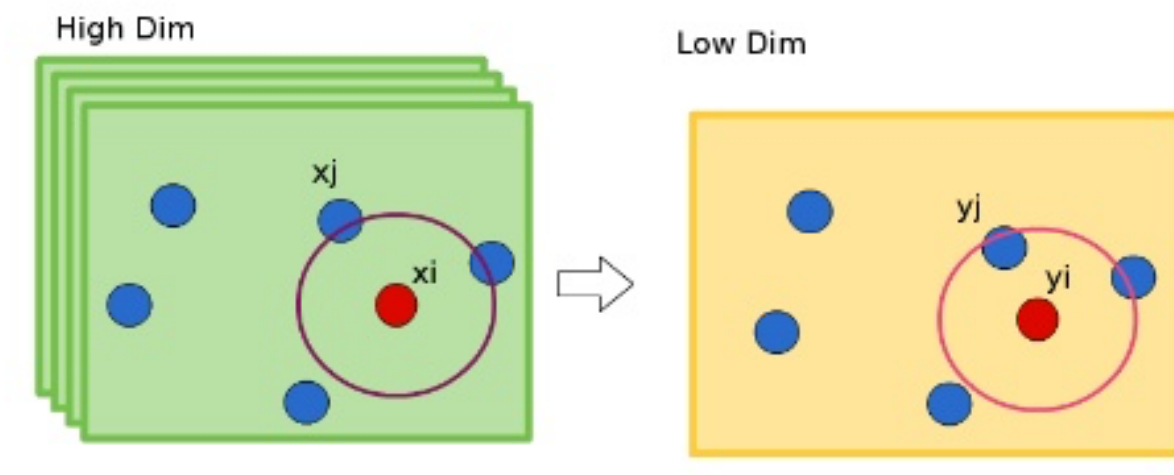
Stochastic Neighbor Embedding (SNE) converts the high-dimensional Euclidean distances between data points into conditional probabilities that represent similarities (using t-Student distribution). Then do it for low-dim.



From https://www.slideshare.net/ssuserb667a8/visualization-data-using-tsne

Create a Similarity Matrix
Then make it better!

Stochastic Neighbor Embedding (SNE) converts the high-dimensional Euclidean distances between data points into conditional probabilities that represent similarities (using t-Student distribution). Then do it for low-dim.



SNE's cost function focuses on retaining the local structure of the data

Minimize the cost function by Gradient Descent

From https://www.slideshare.net/ssuserb667a8/visualization-data-using-tsne

Demo

- How to Use t-SNE Effectively
  http://distill.pub/2016/misread-tsne/
  - 1a. Perplexity (5 to 50 is usually best)
  - 1b. Number of Iterations
  - 2. Cluster Size
    - t-SNE's measure of "distance" varies by local density
    - As a result, it naturally expands dense clusters, and contracts sparse ones, evening out cluster sizes.
  - 3. Cluster Distance
  - 5b. Shapes (2 bars)

Demo (continued)

- Square Grid
  - More points
  - Low perplexity
  - Very high perplexity
- Two Clusters, equal size
  - Very low perplexity => worms

References

- Laurens van der Maaten's t-SNE Github https://lvdmaaten.github.io/tsne/
- In depth presentation by Laurens van der Maaten https://www.youtube.com/watch?v=RJVL80Gg3lA&list=UUtXKDgv1AVoG88PLl8nGXmw#
- Analytics Vidhya blog https://www.analyticsvidhya.com/blog/2017/01/t-sne-implementation-r-python/
- https://en.wikipedia.org/wiki/T-distributed_stochastic_neighbor_embedding
- http://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html
- https://www.slideshare.net/ssuserb667a8/visualization-data-using-tsne

Thank You!