95-865 Lecture 4: t-SNE
(t-distributed stochastic neighbor embedding)

George Chen
t-SNE High-Level Idea #1

- Don't use deterministic definition of which points are neighbors
- Use probabilistic notation instead
t-SNE High-Level Idea #2

- In low-dim. space (e.g., 1D), suppose we just randomly assigned coordinates as a candidate for a low-dimensional representation for A, B, C, D, E (I'll denote them with primes):

- With any such candidate choice, we can define a probability distribution for these low-dimensional points being similar.
t-SNE High-Level Idea #3

- Keep improving low-dimensional representation to make the following two distributions look as closely alike as possible.

This distribution stays fixed.

This distribution changes as we move around low-dim. points.
Technical Detail for t-SNE

Fleshing out high level idea #1

Suppose there are \( n \) high-dimensional points \( x_1, x_2, \ldots, x_n \)

For a specific point \( i \), point \( i \) picks point \( j \) (\( \neq i \)) to be a neighbor with probability:

\[
p_{j|i} = \frac{\exp(-\frac{||x_i-x_j||^2}{2\sigma_i^2})}{\sum_{k\neq i} \exp(-\frac{||x_i-x_k||^2}{2\sigma_i^2})}
\]

\( \sigma_i \) (depends on \( i \)) controls the probability in which point \( j \) would be picked by \( i \) as a neighbor (think about when it gets close to 0 or when it explodes to \( \infty \))

\( \sigma_i \) is controlled by a knob called 'perplexity'
(rough intuition: it is like selecting small vs large neighborhoods for Isomap)

Points \( i \) and \( j \) are "similar" with probability:

\[
p_{i,j} = \frac{p_{j|i} + p_{i|j}}{2n}
\]

This defines the earlier blue distribution
Technical Detail for t-SNE

Fleshing out high level idea #2

Denote the $n$ low-dimensional points as $x_1', x_2', \ldots, x_n'$

Low-dim. points $i$ and $j$ are "similar" with probability:

$$q_{i,j} = \frac{1}{1 + \| x_i' - x_j' \|^2} \sum_{k \neq m} \frac{1}{1 + \| x_k' - x_m' \|^2}$$

This defines the earlier green distribution

Fleshing out high level idea #3

Approximately minimize (with respect to $q_{i,j}$) the following cost:

$$\sum_{i \neq j} p_{i,j} \log \frac{p_{i,j}}{q_{i,j}}$$

This cost is called the “KL divergence” between distributions $p$ and $q$
Manifold Learning with t-SNE

Demo
t-SNE Interpretation

https://distill.pub/2016/misread-tsne/
Visualization is a way of debugging data analysis!

Example: Trying to understand how people interact in a social network

Important:
Handwritten digit demo was a toy example where we know which images correspond to digits 0, 1, … 9

Many real UDA problems:
The data are messy and it’s not obvious what the “correct” labels/answers look like, and “correct” is ambiguous!

This is largely why I am covering “supervised” methods (require labels) after “unsupervised” methods (don’t require labels)

Top right image source: https://bost.ocks.org/mike/miserables/
Dimensionality Reduction for Visualization

• There are *many* methods (I've posted a link on the course webpage to a scikit-learn example using ~10 methods)

• PCA is very well-understood; the new axes can be interpreted

• Nonlinear dimensionality reduction: new axes may not really be all that interpretable (you can scale axes, shift all points, etc)

• PCA and t-SNE are good candidates for methods to try first

• If you have good reason to believe that only certain features matter, of course you could restrict your analysis to those!