UMAP: Uniform Manifold Approximaation and Projection for Dimension Reduction

Shubham Gupta

January 17, 2020

1 Introduction

- Types of dim reduction techniques
  - Preserve the distance structure within the data. *PCA, MDS, Sammon Mapping*
  - Favor presevation of local distances over global distance. *t-SNE, Isomap, LargeVis, Laplacian eigenmaps, diffusion maps*
- Compared to t-SNE, preserves more global structure, better run time performance, scales to larger dataset sizes and no computational restriction on embedding dimension.

2 Theoretical foundation

- Skipping this in the first read. Will come back to it.

3 Computational View of UMAP

- Core idea: *Fuzzy simplical sets* which are generalizations of directed graphs, partially ordered sets and categories.
- UMAP basically consists of construction and operations on weighted graphs

4 Phases

- Two phases
  - Weigted k-neighbour graph is computed. Theoretical explaination in section 2, which I've not yet read lol.
  - Low dimensional layout of this graph is computed

4.1 Graph Construction

-

5 Video

- Find the "latent" features in the data
- Lot of redundant dimensions. Reducing it can improve algorithms
- Techniques
  - Matrix factorization
  - Neighbour graphs
- t-SNE is SOTA in neighbour graph algorithms

5.1 UMAP

- Neighbour graph with maths(lol)
- Topological analysis
  - **Simplicies** Combinatorial representations of topological spaces
  - Can use a combination to build multiple complex spaces
  - **Nerve Theorm**: If we build a simplex out of a topological space in a certain way, we can recover all the important topology
  - Start by building a cover for the input data i.e open balls on each point.
  - Make a simplicial complex using nerve of the cover(not sure if he said nerve)
  - If data is uniformly distributed, the cover will be *good*
  - **Assumption** Data is uniformly distributed on te manifold
  - Define Riemannian metric that makes this assumption true
  - Patches of data mapped down to euclidean space on different spaces
  - Choose fuzzy cover for the manifold
  - **UMAP Adjunction theorm**: Couldn't understand this bit.
  - **Assumption** Assume manifold is locally connected i.e it cannot have isolated points
  - Why this assumption?
    - Increase in dimension, distribution of distances increases
    - Normalize results in tighter bounds i.e no clear distances
    - With local connectivity and normalize, distributions of distances look better.
  - Local metrics are incompatible
  - Parameters $\tau_{\beta,\alpha}$ and $\tau_{\alpha,\beta}$ inform us how to move back and forth between the two projections
  - **Theorm**: Convert everything into fuzzy simplical sets and take the union of these sets to get the final answer
  - Combine weights using formula: $f(\alpha, \beta) = \alpha + \beta - \alpha.\beta$ (looks similar to cosine distance rule I think)

- Weight on an edge is the prob the edge exists
- **Need a low dim rep for this process**
- Apply the same method to get a fuzzy graph
- We know manifold but don't know correct nearest neighbour distance
- Measure distance between two graphs using cross entropy and optimize

$$CE = \sum_{a \epsilon A} \mu(a) log \frac{\mu(a)}{\nu(a)} + (1 - \mu(a)) log \frac{1 - \mu(a)}{1 - \nu(a)} \tag{1}$$

- First term gets the clumps right similar to t-SNE
- Second term gets the gaps right similar to PCA
- Needs
  * Find nearest neighbors fast even with higher dim data
  * *Solution*: RP-trees + NN-descent
  * Optimze the layout subquadratucally
  * *Solution* SGD + negative sampling(similar to word2vec)
  * High level but still fast
  * *Solution* Python + numba

5.2 Next steps

- Embed new unseen points into an existing embedding
- Make use of labels and do supervised dim reduction
- Combine above and do metric learning
- Adding one categorical variable is no harder than adding many others
- Combine spaces with different metrics
- UMAP for pandas dataframes
- Tree sampling paper: Dasgupta +Freund 2018

6 Conclusion

- Video seems interesting. It's cool to see that UMAP has solid theoritical foundations for how it lowers the dimensionality of data, and that it is not just a visual tool.
- I didn't really understand the math because it was too dense to follow, but maybe reading the paper will give me more insights(specifically section 2 of the paper).