Parametric UMAP embeddings for representation and semi-supervised learning - An overview

Jan Ruman

22.5.2021

Abstract

This document attempts to summarize the Parametric UMAP embeddings for representation and semi-supervised learning research paper by Sainburg et al.

1 Introduction

Dimensionality reduction is a problem of mapping high-dimensional data to low-dimensional representations while retaining as much topological structure as possible.

One of the methods attempting to solve this problem is UMAP [2]. However, UMAP is a non-parametric method, and it is thus not very practical to use in an online fashion.

The authors thus propose Parametric UMAP [4] - a UMAP derivative, that uses a parametrical model (a deep neural network) to compute the low-dimensional representations. Moreover, they propose multiple ways the original UMAP method can enhance parametrized models in general.

2 Parametric and non-parametric methods

In general, methods for dimensionality reduction can be divided into two categories - parametric and non-parametric. Some methods combine the two; we shall call those parametric too.

2.1 Parametric methods

Parametric methods make the assumption that the mapping can be described by a function parametrized by set of learnable parameters [1]; this makes prediction on unseen data very easy - all we need to do is to pass the data through a learned function. An example of this approach would be an autoencoder.

2.2 Non-parametric methods

Non-parametric methods learn to map high-dimensional data to their low-dimensional counterparts; however, this mapping is learned only for the given dataset. Even though these methods tend to make fewer assumptions about the data, the problem is that classification of new examples is non-trivial. An example of this approach is UMAP (or t-SNE).

3 Parametric UMAP

To understand how Parametric UMAP works, we need first to understand how UMAP works.

3.1 UMAP

UMAP is a non-parametric dimensionality reduction technique. It assumes the high-dimensional data is uniformly distributed on a low-dimensional manifold, which is ensured by defining a Riemannian metric on the manifold. The distance from x_i to x_j is defined as:

$$p_{j|i} = \exp(\frac{-\max(0, d(x_i, x_j) - \rho_i)}{\sigma_i})$$

where ρ_i is the distance from nearest neighbor of x_i and σ_i is chosen to satisfy the uniformity constraint. The distance from one point to another can be viewed as the probability that an unweighted directed edge exists in a complete graph induced by the dataset. To make the distances symmetric, authors of the UMAP paper propose (and theoretically justify) to compute the distance between x_i and x_j as

$$p_{ij} = (p_{j|i} + p_{i|j}) - p_{j|i}p_{i|j}$$

Intuitively, we can think of the resulting probability of an edge to be the probability that either of the one-directional edges exists.

In the low-dimensional space, the distance between points z_i and z_j is computed as

$$q_{ij} = (1 + a||z_i - z_j||^{2b})^{-1}$$

Where a and b are hyperparameters based upon the desired minimum distance between points in embedding space.

The authors chose cross-entropy as the loss function. Gradient descent with negative sampling is used to optimize the loss.

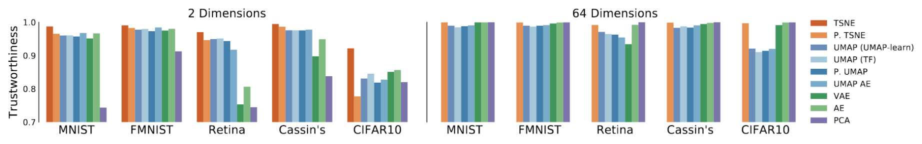

Figure 1: Comparison of individual methods. Taken from the original paper [4]

3.2 Parametric UMAP

Parametrization of UMAP is a relatively trivial procedure - instead of computing the gradient of UMAP loss function with respect to low-dimensional points we use a deep neural network to compute the low-dimensional embeddings from their high-dimensional counterparts and optimize the weights using the UMAP loss.

3.3 Other UMAP parametrizations

There are more ways to utilize UMAP in parametric models. The authors suggest two particular use cases.

To further improve the Parametric UMAP method authors suggest regularizing it using a reconstruction loss, obtained by extending the model to an autoencoder [3].

Another way to utilize UMAP is to use it as a regularization in semi-supervised setting. A network that outputs a low-dimensional embedding of a datapoint and its class can be trained using both UMAP and classification loss. This is particularly useful in semi-supervised scenarios where we only have a small amount of labeled data.

4 Experiments

Because of space limitations, we are not going to cover all the experiments the authors conducted.

4.1 Embeddings

To assess the quality of obtained embeddings, the authors compare their methods with other methods in the field of dimensionality reduction, namely t-SNE, parametric t-SNE, AE, VAE, and PCA. The authors used multiple metrics for the assessment; we are only going to look at one of them - trustworthiness, i.e., how much does a given method preserve the local structure of the data.

In Figure 1, we can see the performance of individual methods on several datasets. The dimensionality reduction was performed either to 2D or 64D. The most important observation is that Parametric UMAP was able to perform on par with regular UMAP. In the 2D case, t-SNE was able to outperform UMAP and its derivatives systematically; PCA severely underperfomed compared to

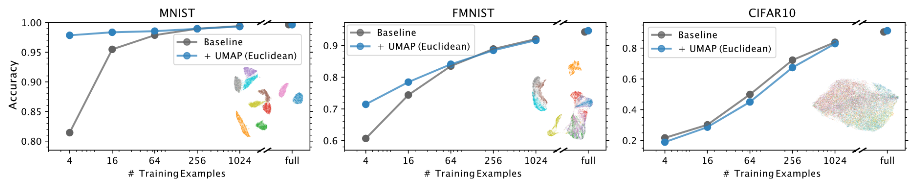

Figure 2: Using UMAP loss to aid semi-supervised learning. Taken from the original paper [4]

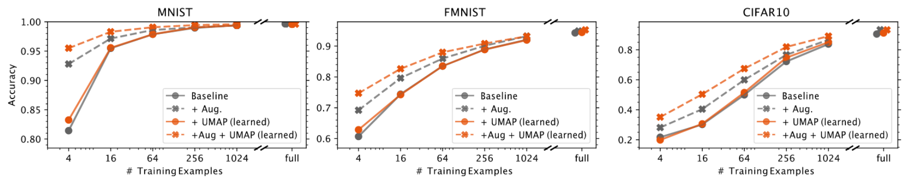

Figure 3: Using UMAP on classification network activations instead. Taken from the original paper [4]

other methods. The most interesting observation in the 64D case is CIFAR-10: UMAP and its derivatives underperformed compared to other methods; moreover, AE performed better than UMAP AE.

4.2 Speed

Parametric UMAP took longer to train than regular UMAP to achieve a given cross-entropy, but the differences were typically within an order of magnitude. As expected, the embedding and reconstruction speeds were orders of magnitude better in the case of parametric methods.

4.3 Autoencoding

The experiment results did not show a clear superiority of UMAP AE over AE.

4.4 Semi-supervised learning

We can utilize both labeled and unlabelled data by jointly training a neural network on both classification and UMAP loss. As can be seen in Figure 2, this approach improved the performance on MNIST and FMNIST when the number of training examples was very small, but it led to worse accuracy in the case of CIFAR-10.

The same results could be observed when using augmented data; in this case UMAP is trained to be invariant to data augmentation.

The authors conjecture that pixel-wise distances do not constitute a good distance metric between CIFAR-10 images; they thus propose to use the activations from the classification network to be used for UMAP. This technique does not improve the accuracy when used alone; however, if used together with data augmentation, it is able to significantly improve the performance when only small number of examples is labeled.

5 Conclusions

The authors proposed a novel technique for dimensionality reduction based on UMAP. Most importantly, they create a parametric method based on UMAP, allowing for faster inference. They also suggest using UMAP in other cases, namely autoencoding and semi-supervised learning. Experimental results show that Parametric UMAP performs on par with UMAP, that autoencoder benefits very little from UMAP loss regularization, and that UMAP loss can be used to enhance the performance of neural networks in a semi-supervised setting.

References

[1] Christopher M Bishop. *Pattern recognition and machine learning*, pages 120–127. springer, 2006.

[2] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[3] David E. Rumelhart and James L. McClelland. *Learning Internal Representations by Error Propagation*, pages 318–362. 1987.

[4] Tim Sainburg, Leland McInnes, and Timothy Q Gentner. Parametric umap: learning embeddings with deep neural networks for representation and semi-supervised learning. *arXiv preprint arXiv:2009.12981*, 2020.