Symmetric Positive-Definite matrixes (SPD) and their applications

Nikita Durasov
Optimization class project, MIPT

Introduction

Symmetric positive definite (SPD) matrices have shown powerful representation abilities of encoding image and video information. In computer vision community, the SPDmatrix representation has been widely employed in many applications, such as face recognition (*e.g.*Pang, Yuan, and Li2008; Huang et al. 2015; Wu et al. 2015; Li et al. 2015), object recognition (Tuzel, Porikli, and Meer 2006; Jaya-sumana et al. 2013; Harandi, Salzmann, and Hartley 2014; Yin et al. 2016), action recognition (Harandi et al. 2016), and visual tracking (Wu et al. 2015).

The SPD matrices form a Riemannian manifold, where the Euclidean distance is no longer a suitable metric. Previous works on analyzing the SPD manifold mainly fall into two categories: the local approximation method and the kernel method.

However, both local approximation and kernel methods face two problems. First, the SPD matrices are high dimensional, which brings the problem of high computational cost. Second, the vectorization operation on SPD matrices might giverise to the distortion of the manifold geometrical structure.

Motivated by achievements of deep networks, we advocate modeling the non-linear mapping which reduces the dimensionality of high dimensional SPD matrices via a deep neural network.

Basic blocks

We introduce two basic layers, i.e., the 2D fully connected layer and the symmetrically clean layer, to realize dimension reductionand non-linear operation, respectively.

- $Y = W^T XW$ - 2D fully connected layer
- Symmetrically Clean Layer - zero set of symmetric elements (matrix is still SPD)

ReLU operation is used here for simplicity. The ReLU operation assigns all the negative elements in the SPD matrix to zero. The principal diagonal elements of an SPD matrix will be positive forever, so the ReLU operation is a suitable choice.

Encoder approach

Our approach is the build coder and decoder for symmetric positive-semidefinite laplacian matrix of graph, consisted of basic block from previous paragraph. Formally, we are going to solve next problem:

$$\text{minimize}_{\theta,\omega} \sum s(\|x - f(g(x|\omega)|\theta)\|)$$

where f and g are decoder and encoder respectively.

The following models for encoding will be considered:

- Simple Autoencoder
- Variational Autoencoder
- Convolutional Autoncoder
- Encoder based on basic blocks

Method

We solve this problem using an Adam and MSELoss

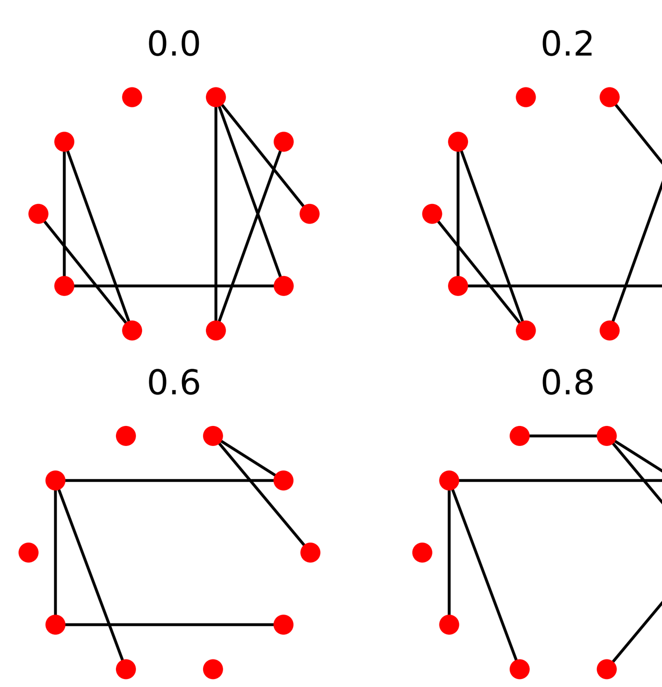
```
encoder_g = {Linear2D, SymmetricallyClean, Tanh}
decoder_f = {Linear2D, SymmetricallyClean, ReLU}
optimizer = Adam()
criterion = MSELoss()

for iter = 1:MAX_ITER
    batch_a = decoder_f(encoder_g(batch))
    loss = criterion(batch_a, batch)
    loss.backward()
    optimizer.step()
end
```

Experiments setup

For experiments we'll use graph datasets MUTAG consisted of ≈ 200 points with two labels and synthetic dataset of SPD matrixes.

After building embeddings for graphs classification problem will be considered on MUTAG. For synthetic dataset there are next problem: using embeddings of SPD matrix predict value of it's determenant (ideally faster than common algorithm).

Example of builded embeddings: decoding embeddings between first and last



Results

Table 1: Min loss value (MSE) for method

Dataset	Simple AE	VAE	Conv AE	BB AE
MUTAG	1.8	-	2.3	**0.04**
Synthetic SPD	103.6	-	102.4	103.2
Determenant	2.6	-	2.4	2.2

Difference between Determenant and Synthetic SPD tasks models of encoder and predictor is in Synthetic SPD they were train separately, but in Determenant it's one model.

It can be seen, that all models failed on Determenant and Synthetic SPD tasks, but on spefic dataset MUTAG matrix model shown significantly better results.

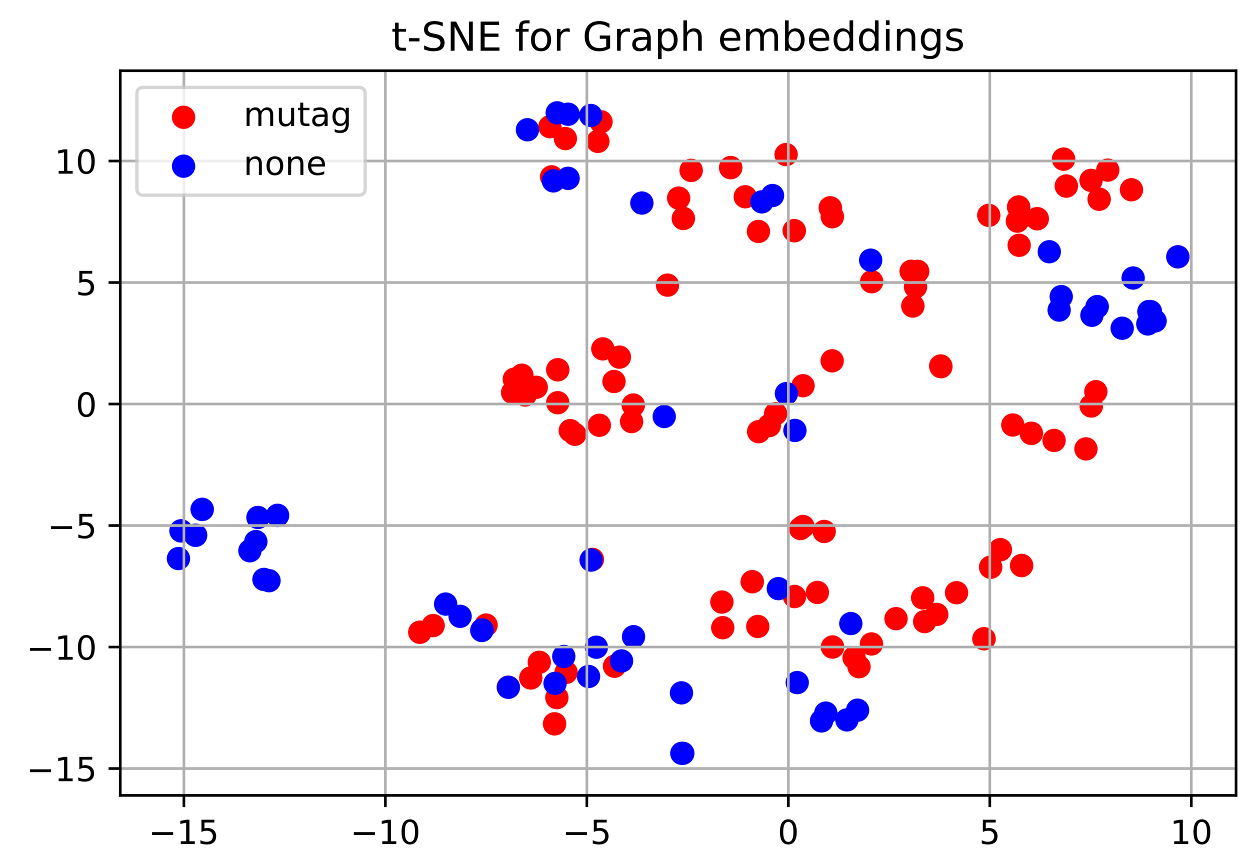


Table 2: Classification results for mutag dataset

	precision	recall	f1-score
none mutag	0.82	0.91	0.86
mutag	0.75	0.56	0.64
avg	0.79	0.8	0.79

Conclusion

We have constructed a deep neural network which projects SSPD Laplacian matrices to a more discriminative low dimensional SPD manifold. Experiments on several graph datasets showed the effectiveness of the proposed network. This technique can be used for constructing low dimensional embeddings for graph structures. Moreover, this approach was tested on SPD matrix in general and it failed.

References:

- Deep Manifold Learning of Symmetric Positive Definite Matrices with Application to Face Recognition
- Manifold Learning Theory and Applications
- graph2vec: Learning Distributed Representations ofGraphs