CS6101 Week 5: Graph Based Models

Lecturers: Christopher Hendra, Mun Kar Kin, Nguyen Van Hoang

Authors: Christopher Hendra, Mun Kar Kin, Nguyen Van Hoang, Hu Hengchang, Guilherme Zagatti

Slide Link:
https://docs.google.com/presentation/d/1hrK0-ixRuTbfg1tLdM58AIZnDwqqripoDXSBwbiu_Z8

This document:
https://docs.google.com/document/d/1mdApQ2WGS0dcs7nK1nRPsJidLfxmPLZ5qtPfiXmtceE
Other documents in this series at http://bit.ly/cs6101-1920-notes

Summary

Geometric Deep Learning

The first half of the lecture was based on the NIPS 2017 tutorial Geometric Deep Learning on Graph and Manifolds (video and article). We started by discussing the reasons behind the effectiveness of convolutional neural networks (CNN) on images. There are three main assumptions behind the structure of images that CNN effective, namely:

1. Shift invariance: the position of objects in an image does not affect the classification task;
2. Locality: the relationship between nearby pixels is more relevant to the classification task than the relationship of pixels that are far apart;
3. Compositionality: an image is composed of hierarchical structures that can be leveraged for the classification task.

CNN leverages these assumptions for implementing learning tasks that make efficient use of convolution filters, pooling and hierarchical layering. CNN achieve the following complexity:

- O(1) parameters per filter;
- O(n) complexity per layer;
- O(log n) layers in classification tasks.

Can we implement similar learning procedures on data such as graphs which are defined on a non-Euclidean space with similar performance?

One of the key differences between Euclidean data and non-Euclidean data is the notion of direction and distance. The notions of up, down, right and left are absent in graphs which are only defined by the relationship between nodes. For the same reason, it is difficult to determine the notion of distance between nodes. Nevertheless, we can intuitively define the notions described above for graph data:

1. Shift-invariance: consistent aggregation of neighbouring features regardless of location;
2. Locality: using proximity sampling to capture features in a local spatial neighborhood;
3. Compositionality: modelling structural hierarchy such as a community.

In order to operationalize these concepts two methods were presented: spectral- and spatial-based methods. Graphs can be represented as two Hilbert spaces with operations defined on nodes and edges representing scalar and vector fields. These representations allow us to define differential operators such as gradients representing functions from nodes to edges and divergence representing functions from edges to nodes. These differential operators can be combined to form a Laplacian which can intuitively be interpreted as the weighted mean divergence for each node applied to any function defined on nodes. In graphs the Laplacian can be interpreted as a matrix consisting of the degrees matrix minus the adjacency matrix. This is a positive semi-definite matrix.

Spectral-based methods

The idea behind spectral-based methods is to project the input features into the spectral sub-spaces obtained by decomposing the Laplacian matrix into orthogonal subspaces of increasing energy. In this case, the energy is the Dirichlet energy which measures how smooth a given function changes along the scalar space. A weights diagonal matrix of spectral multipliers representing a filter in the frequency domain is used to weigh the different subspaces in the spectral domain. The data is then reprojected back to the spatial domain and a nonlinear function is used to obtain an output value.

The key challenge with the spectral-based method presented above is that it is domain specific and the trained model can only be applied to a graph with the same structure. Further to that, this method still takes O(k) parameters per layer and $O(n^2)$. In order to avoid this problem, the weights diagonal matrix can be parametrized with basis functions that can be applied in multiple domains. The main advantages are:

1. O(1) parameters per layer;
2. Filters are guaranteed r-hops support, where r is the order of the approximating polynomial;
3. No explicit computation of the spectral sub-spaces reducing complexity to O(nr).

Among the most popular implementations of spectral-based methods we find ChebNet and Graph Convolutional Networks (Kipf and Welling 2017).

Spatial-based methods

The second family of models to implement neural networks on graphs is the spatial-based methods. In this case we treat each node as a data point and edges represent relation between data points. This family of models attempt to address a problem with spectral-based models that they do not allow for inductive inference, that is, they do not allow for generalization from observed examples. Spectral methods tend to employ transductive inference to generalize on the same target domain. Spatial-based methods look at each node and its surrounding neighbors to draw general rules (induction).

GraphSage is an example of a spatial method that operates directly on the graph structure. It operates in three steps:

1. Sample neighborhood
2. Aggregate feature information from neighbors
3. Predict graph context and label using aggregated information

In order to improve representation learning, GraphSage employs a loss function that not only minimizes the aggregation function around the neighborhood but also penalizes similarity of far away nodes.

This algorithm operates very similar to NLP algorithms. The latter pick words relative to its position in a sentence and its neighbors. GraphSage picks nodes relative to its position in a graph and its neighbors. This analogy gives us some clue on how to employ different aggregate functions common in NLP. GraphSage uses LSTM as an aggregate function. Another spatial-based method, Graph Attention Network (GAT), uses attention cells.

As an application of spatial-based methods, we discussed DialogueGCN which is used to recognize emotions in conversation. The model assumes M speakers and N utterances. It attempts to predict eight labels related to emotions for each utterance.

Graph embeddings

In the graph structure model, each node is a labelled data point, and each edge represents relation between data points.

There are lots of ways to propagate information in graph structure. Our presenter mentioned several traditional ways, including node2vec, which analogy with word2vec. It uses DeepWalk, which is a stochastic process starting at a node. And there are also some other traditional propagation methods that were discussed in the meeting:

1. PageRank
2. Key Phrase Extraction(TextRank)
3. LexRank

One application used by graph embedding model is Emotion Recognition in Conversation. Presenter showed the outperforming of DialogueGCN, compared to earlier works. They utilizes RNN and GRU but they do not always propagate long-term contextual information.

DialogueRNN employs attention mechanism that pools information from entirety or part of the conversation per target utterance, however it does not consider speaker information of the utterances and the relative position of other utterances from the target utterance

- Mentioned SDNE, struc2vec

Resources

1. Introduction to graph base models
    - *NIPS 2017 tutorial, Geometric Deep Learning on Graphs and Manifolds (video and article)* The first half of our tutorial was based on this lecture which provides insightful ideas about non-euclidean machine learning. The paper is math heavy but brings a very nice unifying framework for neural networks. These blogs in Medium (part 1 and part 2) provide a more intuitive idea of the concepts presented in the NIPS lecture.
2. Survey papers
    - Representation Learning on Graphs: Methods and Applications
    - A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications
    - A Comprehensive Survey on Graph Neural Networks
    - Must-read papers on GNN
3. Applications to representation learning
    - Laplacian eigenmaps for dimensionality reduction and data representation spectral embeddings by using eigendecomposition of graph laplacian matrix
    - DeepWalk: online learning of social representations generalizing contextual sampling from word2vec into proximity sampling for node embedding
    - node2vec: scalable feature learning for networks improving DeepWalk with controlled random walk
    - Variational graph auto-encoders a variational auto-encoder application on graph's unsupervised learning
    - LINE: large-scale information network embedding
    - Struc2vec: learning node-representations from structural identity
4. Applications to supervised learning
    - Semi-supervised classification with graph convolutional network efficient approximation of convolution filters on graph.
    - Gat: graph attention networks leveraging self-attention by attending to neighbouring features and not requiring costly matrix operations

- Inductive representation learning on large graphs an inductive representation learning approach
- Neural message passing for quantum chemistry adding edge information and when the information is propagated
- Relational inductive biases, deep learning, and graph networks adding global information of graph during information aggregation

5. Open-source implementations
    - Deep graph library
    - OpenNE: an open source toolkit for network embedding