Distributed Divide & Conquer Techniques for Vertex Classification

Gangal Varun CS11B038
S.K.Ramnandan CS11B061

Introduction

Traditionally, we have looked at classification algorithms to predict class labels for entities based on attributes of the entity alone. In real-world datasets, entities generally form a network and a lot of latent information is hidden in the interlinking of entities which can be used in the prediction of class labels.

Problem Overview

We are given a directed graph $G = (V, E)$ with vertex set V and edge set E. Each vertex has set of attributes associated with it and belongs to one of the several classes. A common way of representing this is to associate a feature vector with every vertex. The task at hand is to assign a class label to unlabeled vertices in the graph. This problem is commonly known as the Vertex Classification Problem.

Since real-world datasets are huge, we look at distributed divide-and-conquer approaches to classify vertices of the graph. We aim to analyse speed-vs-quality of different divide and conquer approaches and attempt to devise an approach tailored to the problem of vertex classification.

To better illustrate this, here is one of the problems we wish to study. A community (also referred to as a cluster) is a set of cohesive vertices that have more connections inside the set than outside. A common approach while analysing graphical models is to *divide* the graph into communities. Since we are interested in the class label of a vertex, we allow the communities to be overlapping. The fringe elements (vertices which belong to more than one community) can play the role of mediators between different communities.

We would look to develop a distributed implementation of overlapping community detection and then, once the graph is divided into overlapping communities and classification is done within, we need to *merge* the results of the *divide* and *conquer* phases, which is often the most difficult phase. Even a distributed implementation of overlapping community detection would be time-consuming and may not provide a dramatic improvement in quality of classification over a random partitioning of the graph in the *divide* phase. This is an example of an approach we would be analyzing. We wish to employ and experiment techniques such as PageRank to determine the*importance* of a vertex in the classification of its neighbours.

During the conquer phase of the algorithm, we can use the feature vectors which are associated with each vertex for classification, in addition to the underlying graph structure.

After having done a thorough analysis of approaches, we wish to develop novel strategies for the *divide, conquer* and *merge* phases tailored specifically for the problem of vertex classification.

Datasets

We plan to perform the experiments on the following real-world graphs. Of particular interest are labeled, directed and connected graphs.

Citation Networks

In a citation network, vertices indicate research papers. An edge from one vertex to another indicates a citation of the former paper by the latter one.

1. **Document Classification:** In the following datasets, each vertex is attributed with a 0-1 word vector indicating the presence of the corresponding word from the dictionary. Our task is to classify each paper as belonging to a particular research area (such as Artificial Intelligence or Machine Learning). The set of research topics(class labels) is predetermined.

   (a) **CiteSeer**: The CiteSeer dataset consists of 3312 scientific publications classified into one of six classes. The citation network consists of 4732 links. The dictionary consists of 3703 unique words

   (b) **Cora**: The Cora dataset consists of 2708 scientific publications classified into one of seven classes. The citation network consists of 5429 links. The dictionary consists of 1433 unique words.

   (c) **WebKB**: The WebKB dataset consists of 877 scientific publications classified into one of five classes. The citation network consists of 1608 links. The dictionary consists of 1703 unique words.

2. **NBER Patents**: The National Bureau of Economic Research provides a citation graph which includes all citations made by 3,923,922 patents granted between 1975 and 1999, totaling 16,522,438 citations. Each patent is classified into one of 987 classes. Each patent has a set of attributes associated with it such as state of author, measure of originality, etc.

3. **cit-HepTh**: Arxiv HEP-TH (high energy physics theory) citation graph is from the e-print arXiv and covers all the citations within a dataset of 27,770 papers with 352,807 edges. Each vertex(paper) has an associated meta-information file describing it, which we will convert into a feature vector.

Social Networks

In the case of social networks, it is not so easy to come across labeled datasets but based on the metadata associated with vertices and edges , we can annotate the vertices based on the domain of interest. For instance, using a user's employment and salary information we can annotate whether the income group is low, medium or high.This can then be approached as a vertex classification problem as in the case of citation networks.

- **Twitter**: This dataset consists of circles (or lists) from Twitter. Twitter data was crawled from public sources. The dataset includes node features, circles, and ego networks.It contains 81306 nodes and 1768149 edges.

- **Google Plus**: This dataset consists of circles from Google+. Google+ data was collected from users who had manually shared their circles using the 'share circle' feature. The dataset includes node features (profiles), circles, and ego networks.It contains 107614 nodes and 13673453 edges.

Image Repositories

Image repositories are platforms where users can upload, share and tag images. Vertices represent images and an edge represents various associations between them such as sharing the same tag or having the same location. The task of image classification in this setting can be reduced to a vertex classification problem by annotating images based on the meta-data. For instance, we can annotate whether images are person or non-person images.

- **Flickr**:This dataset is built by forming links between images sharing common metadata from Flickr. Edges are formed between images from the same location, submitted to the same gallery, group, or set, images sharing common tags, images taken by friends, etc. It contains 105938 vertices and 2316948 edges.

Evaluation of Outcome

We have carefully chosen datasets which provide true class labels for the vertices of the graph. Since we are interested essentially in classification of vertices, we split the dataset into training and test datasets and compare the predicted class labels of the vertices in the test datasets against the true class labels.The F-measures are a commonly used family of measures used to determine the goodness of classification. Let us denote the set of class labels as G, the set of vertices classified to the i^{th} class as C_i and the set of vertices truly belonging to the i^{th} class as S_i . In general,the F_β measure is defined as follows:

$$F_\beta(i) = (1+\beta^2)\frac{precision(i).recall(i)}{\beta^2.precision(i) + recall(i)} \tag{1}$$

where β is a non-negative real value and the precision and recall of S_i are defined as follows:

$$precision(i) = \frac{|C_i \cap S_i|}{|C_i|} \tag{2}$$

$$recall(i) = \frac{|C_i \cap S_i|}{|S_i|} \tag{3}$$

Then,the average F_β measure is defined to be:-

$$\bar{F_\beta} = \frac{\sum_{i \in G} F_\beta(i)}{|G|} \tag{4}$$

Given a set of vertices predicted to be in a particular class, precision indicates how many of those vertices are truly in the same class. Given the set of vertices belonging truly to a particular class, recall indicates how many of those vertices are classified correctly. By definition, the precision and the recall are evenly weighted in F1 measure. On the other hand, the F2 measure puts more emphasis on recall than precision. It is important to quantify the recall in cases where the vertices in these datasets are partially annotated, i.e., some vertices are not annotated to be a part of a class even though they actually belong to that class. This indicates that it would be reasonable to weight recall higher than precision, which is done by the F2 measure.

The F1 and F2 measures as explained above represent the quality of vertex classification. At the same time, since we are looking at distributed divide and conquer techniques, we will be looking at the time taken by different approaches empirically and devise metrics to sufficiently capture the tradeoff between speed and quality of the different approaches.

References

[1] Whang, Gleich and Dhillon - *"Overlapping Community Detection Using Seed Set Expansion"*

[2] S.Fortunato - *"Community detection in Graphs"*

[3] Stanford Network Analysis Project - **http://snap.stanford.edu/**

[4] Zhou, Lei and Ye - *Large-Scale Dataset Clustering Based on MapReduce and Hadoop*

[5] Rajaraman, Leskovec and Ullman - *Mining Massive Datsets*

[6] Hastie, Tibshitani and Friedman- *The Elements of Statistical Learning*