CS224W Project Proposal: Categorization of Wikipedia Articles

Jean Feng, Chuan Yu Foo, Yifan Mai

1 Introduction

The automatic categorization of various entities, particularly textual documents, is often approached from a classification and natural language processing (NLP) perspective where both nonlinguistic and linguistic features specific to each document are used as inputs to a classifier which categorizes each document in isolation (e.g. see [1]). However, this approach neglects the fact that entities (documents) are often embedded in a network structure which relates them to one another (for instance through hyperlinks or citations). This network can be a valuable source of information and inform the prediction of document categories.

In this paper, we consider the problem of categorizing Wikipedia articles using both article and network features. We first examine and discuss a number of papers on the topic before discussing our proposal in greater detail. Note that some of the papers discussed deal with the link prediction problem instead of the entity categorization problem. However, an entity classification problem can be converted into a link prediction problem as follows. For each category, create a corresponding category node. The task of categorizing an entity can then be transformed ot the task of predicting a link between the entity and a category node.

2 Literature review

2.1 Link-based classification (Lu & Getoor, 2003)

In [2], Lu and Getoor present a model for predicting entity categories using both information about the entities themselves as well as information about the links. Specifically, for each entity, they compute three kinds of link features that summarise the categories of its neighbours: (i) the mode-link, the mode of its neighbours categories (ii) the count-link, a vector containing, for each category, the number of neighbours with that category (iii) the binary-link, a vector containing, for each category, whether or not the entity has a neighbour with that category.

To compute the categories for all the entities in the graph, they use an iterative algorithm in which in each iteration, the link features for the nodes are computed, and the categories are predicted using multiclass logistic regression on the entity and link features. This process is repeated until there is no change in the categories.

Lu and Getoor test their model on three datasets, and show that their model outperforms a baseline classifier which uses only entity-specific features, accomplishing significantly higher F1 scores.

2.2 Hierarchical structure and the prediction of missing links in networks (Clauset, Moore & Newman, 2008)

In [3], Clauset et al. present a model for predicting a hierarchical structure over a given a network and using this predicted hierarchy to then predict missing edges in the network. The primary assumption of the paper is that if the lowest common ancestor of two nodes is close to the two nodes, the nodes will be more likely to have an edge linking the two, and Clauset uses a parameter p_1 to capture this assortative characteristic of the hierarchy. On the other hand, if two nodes have a lowest common ancestor that is very high up in the hierarchy, the nodes will tend to not have a link between the two, which is captured via a separate parameter p_2, called the dissasortative characteristic. The method combines a maximum likelihood approach with a Monte Carlo sampling algorithm on the space of all possible dendrograms.

2.3 Link prediction in relational data (Taskar, Wong, Abbeel & Koller, 2003)

In [4], Taskar et al. propose the use of relational Markov networks (RMNs) in the problem of link presence and link type prediction. In essence, a relational Markov network is a Markov network in factors are defined based on relations - specifically, for each specified relation, for each set of nodes satisfying the relation, a factor is defined, and all such factors for this particular relation share the same parameters. In the paper, Taskar et al. propose two relations they believe to be useful: (i) similarity: if Y and Z are linked from X in the same context, then the link types of X-Y and X-Z should be related (ii) transitivity: the presence and type of X-Y and Y-Z links should be related to the presence and type of X-Z links.

For best classification results, Taskar et al. set up their model as a conditional random field. Learning was done via gradient ascent and belief propagation. To make learning tractable, Taskar et al. reduce the number of candidate links by pre-selecting a moderate number of links as possible candidates in their datasets. Running their model on university webpage data, Taskar et al. show that their model outperforms the baseline of multinomial logistic regression on individual nodes without using link data, accomplishing accuracy gains of 2 - 5 percentage points.

2.4 Iterative classification in relational data (Neville & Jensen, 2000)

In [5], Neville and Jensen propose a method for the categorization of entities in a relational graph that takes advantage of the network structure. The proposed method is an iterative algorithm that alternates between categorizing nodes and updating node features. In the categorization step, the algorithm uses a Bayesian classifier to infer the category of each uncategorized node. It then retains the top k inferences with the highest confidence score. In the update step, the algorithm updates the features of each node - specifically, relational features, which depend on the categories of neighbors are updated to reflect the new categories. The algorithm iterates with increasing k until all nodes have been categorized.

The model was tested on a dataset from the US Securities Exchange Commission on the task of predicting if a given central company was a chemical company or a bank. Four features were used in the Bayesian classifier: the state of incorporation, the number of subsidiaries, whether the company is linked to more than one chemical company through its board members and whether the company is linked to more than one chemical company through its insider owners. Neville and Jensen found that this method produced accurate results for the top 90% most confident predictions, but poor results for the remaining 10%. Unfortunately, as the authors did not provide a baseline for comparison, this result is difficult to evaluate in context.

3 Discussion

One of the chief weaknesses of both the papers on classifying entities (i.e. [2] and [5] is that they assume that each entity belongs only to a single category, and that the categories are disjoint.

However, this assumption is rarely true in most real world networks, including Wikipedia - in Wikipedia, for instance, each page may belong to one or more categories, and categories may include other categories or overlap with other categories. While it may be possible to work around this issue by doing the classification on a per-category basis (i.e. classify whether a node is in category X or not in category X, repeating for all categories X), this fails to capture the relationships between categories. Indeed, it seems that simultaneous understanding of the structure of both the category graph and the entity graph would lead to better classification results. One way to accomplish this is by modelling both categories and entities (and their associated relations) in the same graph.

Another trend we noticed in the papers was that the methods proposed were either easy to learn but insufficiently expressive, or highly expressive but difficult to learn. For instance, while the iterative classification schemes proposed in [2] and [5] are easy to train and scale well to large and densely connected networks, they both use only a small set of local network and link features. In contrast, while the relational Markov networks used in [4] allow the category of a node to be influenced by the category and entity features of far-away nodes as well as more global network characteristics, they are very difficult to train, lack convergence guarantees, and do not scale to large networks without manual filtering due to the quadratic increase in the number of link factors.

It would be nice if we combine both ease of learning and expressivity in a single algorithm which is both easy to learn, scales to large graphs, but which captures global network structure and long-range node dependencies. Since entities are frequently tightly clustered around categories, which are also clustered near related categories, one way to capture longer range category and article dependencies via local connections in the graph is to model both categories and entities in the same graph. Another possibility is define the network structure in such a way that local connections reflect these long range dependencies, an idea we elaborate on in the project proposal.

One final weakness we observed was that of the two papers that used an iterative classification scheme (i.e. [2] and [5]), neither provided a sound theoretical justification for their iterative algorithm (say as maximizing some likelihood function or some other well-defined optimization objective that is correlated with categorization accuracy).

4 Project proposal

4.1 Introduction

Wikipedia is a free online encyclopedia containing over 4 000 000 user-created articles. Each of these articles can be categorized into a small subset of over 800 000 user-created categories ranging from the very general to the very specific, from "matter" and "thing" to "domesticated animals", "dogs" and "robotic dogs". While the set of Wikipedia categories is rich and interconnected, one of the biggest obstacles to these categories becoming useful is the sparse and inadequate categorization of the pages themselves. More specifically, since articles are manually categorized by editors, many articles have missing categories, incorrect categories, or categories that are too general or too specific.

Machine-assisted categorization can be used to perform categorization more efficiently and accurately. There have been a few attempts at automatically categorizing Wikipedia articles, but most previous attempts (e.g. [1]) have focused mainly on using the article text as a means of categorization. However, this emphasis on examining each article's text in isolation neglects the fact that Wikipedia articles and their inter-links form a network, which provides a rich source of information that is useful for making inferences. For instance, if an article A links to 10 other articles of category X, this information would strongly suggest that A should belong to category X as well. Motivated by this observation, we will consider the problem of automatically categorizing Wikipedia articles with the help of both textual and network features.

4.2 Data and problem

For this project, we will be working with the DBpedia dataset, which consists of a processed version of the Wikipedia article dataset that has been reorganized into a more relational, structured, and computer-readable network format. In addition to raw data on 3 000 000 Wikipedia pages and 600 000 categories (such as the text of the pages, links between pages, categories of the pages, links between categories, and so on), the DBPedia dataset also includes a relational description format (RDF) description of a subset of Wikipedia pages and a manually-curated ontology of 359 categories.

We consider the problem of automatically categorizing Wikipedia articles, i.e. predicting the category of an article given the categories of its neighbours and the structure of the category hierarchy. In our prediction task, we will select a subset of test articles, remove their labels, and attempt to reconstruct the missing categories using our predictive model.

4.3 Network Representation

One critical design decision is how we choose to represent DBpedia as a network. This affects the graph algorithms that we can use, and the information that is available to the classification algorithm.

One possibility is to model both articles and categories as vertices in a directed graph. The the categories of various articles would be represented as edges between articles and their respective categories. In addition, edges between articles would represent hyperlinks, and links between categories would indicate that one category is a subcategory of the other. This model makes it easier to take advantage of the structure of the category hierarchy. This model would allow us to use basic graph search methods to perform operations such as finding the category of a node, finding nodes that belong to the category, finding parent categories of a category, or finding sub-categories of a category.

Another possibility is to represent the categories of nodes as a flat feature. In this model, only articles will be represented as vertices in the graph. Categories will instead be represented as properties of the vertices. Edges in the graph can be used to represent various relations between the articles, as defined in DBpedia. For instance, we could link two articles if they are related by a DBpedia relation, such as "is-a-city-of" or "is-producer-of". Under this model, because of the large diversity of properties and relations defined in DBpedia, we should select a subset of these relations, and consider only to the sub-network in which these relations are defined. This model is richer and provides us with more information regarding the links, which can be used to improve classification. However, this might result in an algorithm that is hand-engineered to use the available relations. Such an algorithm would be difficult to generalize to other situations where different kinds of relations are present.

4.4 Predicting article categories

We intend to explore a number of different approaches for category prediction.

One approach is to assume that the articles were generated from a hierarchy of categories and subcategories, which would be similar to the method used by Clauset. As they did, we could generate use a random hierarchical model to generate the set of most likely dendrograms that could represent the category hierarchy. From the most likely set, we can generate a consensus dendrogram (the process is specified in the paper by Clauset). From the consensus dendrogram, we can take all categories k edges away from the root node to be a final set of predicted categories.

Another approach is to use clustering. The general idea is to define a metric on the space of categories and the space of articles, so that articles close in article space should also be close in category space. There are a number of possible metrics we could use. For instance, for category metrics, we could map each article to a feature vector with a dimension for each category, and a 1 in the kth dimension if the article belongs to the kth category and a 0 otherwise, and use the Euclidean metric on these vectors. We could also compute the distance between two articles based on the category tree distance between some subset of their categories. For article metrics we could use the shortest distance in the article graph, or we could also map each article to a feature vector with features such as in-degree, out-degree, neighbor connectivity, text features and so on, and take the Euclidean distance between these vectors.

More sophisticated approaches involve the use of probabilistic graphical models, as in [4]. One straightforward method is to use a Markov network, modelling the top few category labels of an article as latent variables and defining pairwise factors relating the categories of articles that are linked in the article graph. Larger factors that capture more complex relations such as similarity and transitivity, as suggested by [4], may also be introduced to increase the model's expressiveness. The issue with this approach, as noted in the criticism of [4] is that the resulting Markov network will be too dense and contain too many parameters, making learning difficult. As in [4], we could work around by using a restricted dataset with fewer articles and categories, or to otherwise limit the pairs of articles over which the factors are defined.

4.5 Evaluation

We wish to choose an evaluation method that avoids the following problem. In the DBpedia dataset, many articles are missing appropriate categories because they have not yet been added by human editors. In other words, we consider the dataset to be a corrupted version of ground truth, in which many appropriate categorization links are deleted. It is not appropriate to naively consider DBpedia to be the groud truth when comparing our predictions with DBpedia. In particular, we would like to avoid penalizing our algorithm for predicting an edge that is not present in the DBpedia dataset. It is possible that the predicted categorization is appropriate, but missing due to human oversight.

There are two approaches that avoid this problem. As some articles in DBpedia are known to be better curated that others, one approach is to select a subset of articles that we believe to have high-quality categorizations. We can then assume that the categorizations are complete, and simply use it as the ground truth. Within this method, we can sort all predicted categorizations by confidence level, choose a cut-off confidence level as the decision criterion, and calculate the sensitivity and specificity of our predictions against ground truth. We can then vary the decision criterion and plot the ROC curve of the predictive algorithm.

Another approach is to use the assumption that for each entity, the given categorizations in the DBpedia dataset will always have the highest confidence values of all possible appropriate categorizations. In other words, we assume that DBpedia editors add categories by adding the most significant category first, and then add categories in descending order of significance. Thus, any categories that were missed by editors must have been less significant than all added categories. Given this assumption, we would be only interested in testing if our models can identify these top k categories as annotated by the editors. We evaluate the models by taking the k top predictions for each entity, and counting how many of the k categories in the original DBpedia dataset are present. Given that j out of the k top predicted categories are present in the dataset, we would then say that the model has an accuracy of $\frac{j}{k}$ for that particular entity.

References

[1] Gantner, Z. & Schmidt-Thieme, L. (2009). Automatic content-based categorization of Wikipedia articles. *Proceedings of the 2009 Workshop on the Peoples Web Meets NLP*, pp. 32 - 37.

[2] Lu, Q. & Getoor, L. (2003). Link-based classification. *Proceedings of the Twentieth International Conference on Machine Learning.*

[3] Clauset, A., Moore, C. & Newman, M. E. (2008). Hierarchical structure and the prediction of missing links in networks *Nature.*

[4] Taskar, B., Wong, M., Abbeel, P. & Koller, D. (2003). Link prediction in relational data *Advances in Neural Information Processing Systems* 17.

[5] Neville, J. & Jensen, D. (2000). Iterative classification in relational data *Proceedings of the AAAI 2000 Workshop Learning Statistical Models*, pp. 42 - 49.