Deep Reinforcement Learning for Information Retrieval

Contents

1 Background 2
- 1.1 Focused Crawlers 2
- 1.2 Reinforcement Learning 2

2 Using Reinforcement Learning to Spider the Web Efficiently 4

3 Focused Crawling using Temporal Difference-Learning 5

4 Deep Reinforcement Learning With an Action Space Defined by Natural Language 6

5 WebNav : A New Large-Scale Task for Natural Language based Sequential Decision Making 7

References 7

1

Background

1.1 Focused Crawlers

Focused crawlers aim to search and retrieve only the subset of the world-wide web that associates to a specific topic of relevance. The idea is similar to a traditional crawler but every retrieved page is analysed to check if it belongs to the topic, usually, with the help of a content-based web page classifier [1]. If the page belongs to the topic, all its links become new seeds for the crawler. Otherwise, the page is judged not relevant and discarded.

The ideal focused crawler retrieves the maximal set of relevant pages while simultaneously traversing the minimal number of irrelevant documents on the web. Focused crawlers therefore offer a potential solution to the currency problem by allowing for standard exhaustive crawls to be supplemented by focused crawls for categories where content changes quickly. Focused crawlers are also well suited to efficiently generate indices for niche search engines maintained by portals and user groups, where limited bandwidth and storage space are the norm. Finally, due to the limited resources used by a good focused crawler, users are already using personal PC based implementations. Ultimately simple focused crawlers could become the method of choice for users to perform comprehensive searches of web-related materials.

The aim of Focused Crawling is to able to recognize promising links early on, so as to follow the right path. One major issue in focused crawling is that of properly assigning credit to all pages along a crawl route that yields a highly relevant document. In the absence of a reliable credit assignment strategy, focused crawlers suffer from a limited ability to sacrifice short term document retrieval gains in the interest of better overall crawl performance. In particular, existing crawlers still fall short in learning strategies where topically relevant documents are found by following off-topic pages.

1.2 Reinforcement Learning

Determining the next best thing to do under an evolving knowledge of the real world is the focus of reinforcement learning. Its objective is to maximize the cumulative reward obtained when performing some actions, each action leading to an individual reward and to a new state defined in a stochastic manner. Markov decision processes(MDPs) are a common model for reinforcement learning scenarios, where each action leads to a new state and a given reward according to a probability distribution that must be learned. This implies an inherent tradeoff between exploration (trying out new actions leading to new states and to potentially high rewards) and exploitation (performing actions already known to yield high rewards). Although the use of MDPs for modeling data cleaning tasks has been established [2], where several many challenges to using MDPs for data management:

- the state space is typically huge, representing all possible partial knowledge on the data
- states have complex structures, namely that of the data
- rewards are typically delayed, as queries may only be answerable after a long sequence of actions.

2

Using Reinforcement Learning to Spider the Web Efficiently

A major problem faced by the crawlers is that it is increasingly arduous to learn that some sets of off-topic documents often lead reliably to highly relevant documents.

Consider the example of a task looking for researchers working in the field of on neural networks. A large number of these papers are found on the home pages of researchers at computer science departments at universities. When a crawler hits the home page of a university, a good strategy would be to follow the path to the CS department, then to the researchers pages, even though the university and CS department pages in general would have low relevancy scores. While an adaptive focused crawler described above could in principle learn this strategy, it is doubtful that the crawler would ever explore such a path in the first place, especially as the length of the path to be traversed increases.

To explicitly address this problem, Rennie and McCallum used reinforcement learning to train a crawler on specified example web sites containing target documents. Reinforcement Learning permits the credit assignemnt during the search process and hence, allowing the off-topic pages to be included in the search path. The web site or server on which the document appears is repeatedly crawled to learn how to construct optimized paths to the target documents. The approach learns the context within which target documents are located from a small set of web sites, but in principle can back crawl a significant fraction of the whole web starting at each seed or on-topic document. Furthermore, the approach is inefficient in initialization, since the context is constructed by directly branching out from the good set of documents to model the parents, siblings and children of the seed set.

The technique trained a learner with features collected from paths leading up to relevant nodes rather than relevant nodes alone. Such paths may be collected by following backlinks.

One of the main shortcomings of this approach is that it requires a lot of training examples and the method cannot be trained online. This invovles learning to train a crawler on specified example web sites containing target documents. However, this approach puts burden on the user to specify representative web sites.

3

Focused Crawling using Temporal Difference-Learning

The paper discusses how TD learning can be leveraged for link prediction in focused crawling and presents initial evaluations on a confined dataset. In their approach, every web page is represented as a set of 500 binary values and the state of each page is determined by Temporal Difference Learning, in order to minimize the state space. The relevance of a page depends on the set of keywords present in a page.

Neural Networks are used to estimate values of the different stages. During training session, the crawler randomly follows pages for a defined number of steps or until it reaches a relevant page. Each step represents the implementation of action a thus moving the agen from state s_t to s_{t+1}. The respective reward r_{t+1} and the features of the state s_t are used as input to the neural network which is tuned to evaluate the state's potential of belonging to the right path.

During the crawling mode, the crawler maintains a priority list of links to be followed, the priorities are computed by the neural network. Since it is ineffective to download the children pages of the current page the crawler is at, the state value of a children page is inherited by the value of its parent (the current page) or by the average value of its parents, in case the page is pointed by multiple pages.

4

Deep Reinforcement Learning With an Action Space Defined by Natural Language

5

WebNav : A New Large-Scale Task for Natural Language based Sequential Decision Making

Bibliography

[1] Inma Hernández, Hassan A Sleiman, David Ruiz, and Rafael Corchuelo. A conceptual framework for efficient web crawling in virtual integration contexts. In *Web Information Systems and Mining*, pages 282–291. Springer, 2011.

[2] Michael Benedikt, Philip Bohannon, and Glenn Bruns. Data cleaning for decision support.