Reinforcement Learning
Lecture 18a

Gillian Hayes

7th March 2007

School of informatics

Focussed Web Crawling Using RL

- Searching web for pages relevant to a specific subject
- No organised directory of web pages

Web Crawling: start at one root page, follow links to other pages, follow their links to further pages, etc.

Focussed Web Crawling: specific topic. Find maximum set of relevant pages having traversed minimum number of irrelevant pages.

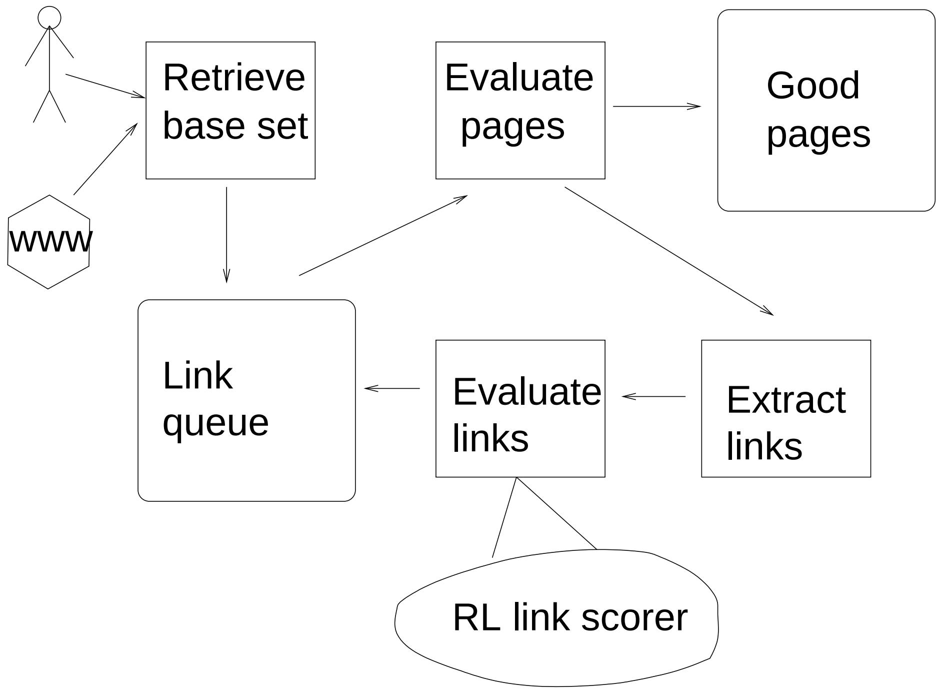
Why try this?: Less bandwidth, storage time (can take weeks for exhaustive search – billions of web pages)

Good for dynamic content – can do frequent updates

Can get indexing for a particular topic

Alexandros Grigoriadis, MSc AI, Edinburgh 2003 + CROSSMARC project – extracting multilingual info from web on specific domains e.g. laptop retail info, job adverts on companies' web pages

Web Crawler



• Link Queue: current set of links that have to be visited. Fetch link with highest score on queue

- Evaluate page this link points to: based on set of text/content attributes. If relevant, store on Good Pages
- Get links from page
- Evaluate links, add to link queue. Does does the link point to a relevant page? will it lead to relevant pages in future?

- Where can we use RL? In the link scorer

RL Crawling

- Reward when it finds relevant pages
- Needs to recognise important attributes and follow most promising links first
- Aim is to get π^*
- How to formulate problem? What are states? What are actions?

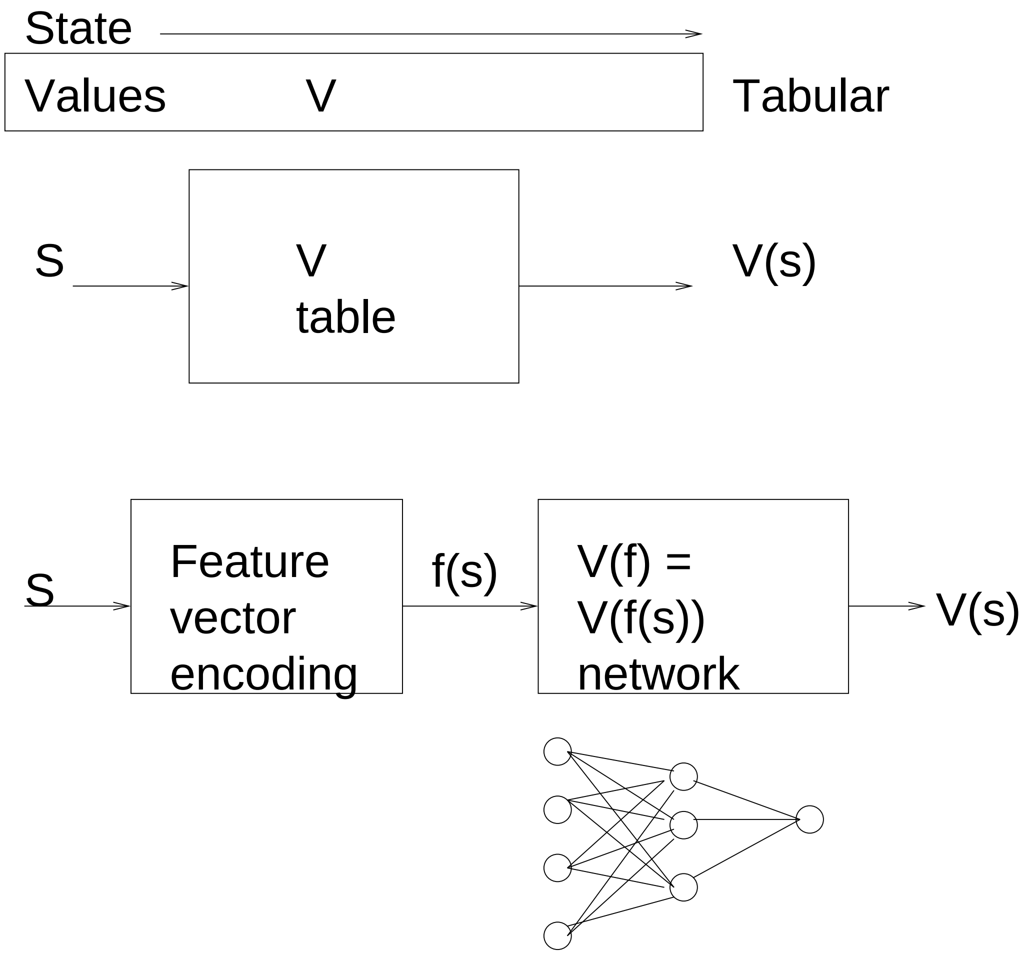
Alternatives:

- State = a link, Action = {follow, don't follow}
- State = web page, Action = links
- Learn V? Must do local search to get policy
- Learn Q? More training examples needed since Q(s,a). But faster to use

Choice: Action–links and learn V using TD(λ)

How to Characterise a State?

- Use text analyser to come up with keywords for domain – these words typically appear on web pages on this subject area
- Feature vector of 500 binary attributes: existence or not of a keyword
- State space: 2^{500} states $\sim 10^{150}$ – too large for a table
- Use a neural network for function approximation to give V(s)
- Learn weights of network using temporal difference learning
- Eligibility trace on weights instead of states
- Reward is 1/0 if page is/is not relevant


State
Values
V
Tabular
S
V
table
V(s)
S
Feature
vector
encoding
f(s)
V(f) =
V(f(s))
network
V(s)

Learning Procedure

- Use a number of training sets of web pages, e.g. different companies' web sites containing numbers of pages with job adverts and start with a random policy
- Learn V^{π}, need to do GPI to get V^{*}
- Then incorporate into a regular crawler: the RL neural net evaluates each page – the V value is its score
- Which link to choose? Must do one-step lookahead – follow all links in current page, evaluate the pages they lead to
- Place new pages on link queue according to score
- Follow link at front of link queue to next page with highest likely relevance

Performance: Finds relevant pages (if $>$1) following fewer links but searches more pages in the 1-step lookahead vs. CROSSMARC non-RL web crawler. Not so good at finding a single relevant page on a site.

• Datasets: up to 2000 pages, 16000 links, tiny number of relevant pages in each dataset, English and Greek, 1000 training episodes

Issues

Depends on: graphical structure of pages

- Features chosen: many attributes were $== 0$ so not discriminating enough
- Need to try on bigger datasets
- Paper outlines alternative learning procedures

Andrew McCallum's CORA – searching computer science research papers

- Treated roughly as a bandit problem learning Q(a). Action a $=$ link on a web page and words in its neighbourhood
- Choose the link expected to give highest future discounted reward
- 53,000 documents, half a million links, 3x increase in efficiency (no. links followed before 75% of docs found vs. breadth-first search)

Alexandros Grigoriadis, Georgios Paliouras: Focused crawling using temporal difference-learning. Proceedings of the Panhellenic Conference in Artificial Intelligence (SETN), Lecture Notes in Artificial Intelligence 3025, 142–153, Springer-Verlag, 2004.

Andrew McCallum et al.: Building domain-specific search engines with ML techniques. Proc AAAI-99 Spring Symposium on Intelligent Agents in Cyberspace