Introduction to Statistical Machine Learning

Christfried Webers

Statistical Machine Learning Group
NICTA
and
College of Engineering and Computer Science
The Australian National University

Canberra
February – June 2013

(Many figures from C. M. Bishop, "Pattern Recognition and Machine Learning")

Outlines

Part XXIV

Selected Topics

Selected Topics

- Occam's Razor
- Reinforcement Learning
- PageRank
- Envelope Paradox

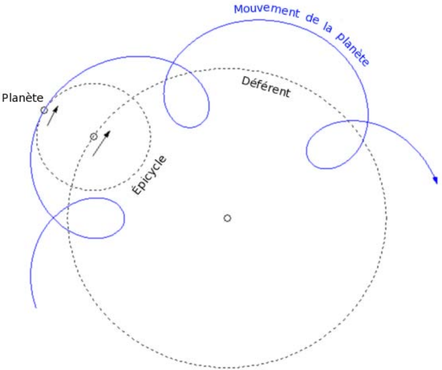
Occam's Razor

- Is there a unique principle which allows to formally arrive at predictions which
  - coincides (always?) with intuitive guesses or better?
  - which is (in some sense) most likely the best or correct answer?
- Occam's Razor : Use the simplest explanation which is consistent with the past data (and apply it for prediction).

William of Ockham (1288 - 1348): "entities should not be multiplied unnecessarily."
He also produced significant works on logic, physics, and theology, e.g. tertiary logic.

Example: Celestial Mechanics

- Why are the planets rotating around the sun?
- And is it ‘true’ anyway?



Occam's Razor

- Occam's Razor : Use the simplest explanation which is consistent with the past data (and apply it for prediction).
- Occam's razor can serve as a foundation of machine learning in general, and is even a fundamental principle (or maybe even the mere definition) of science.
- Karl Popper: a hypothesis, proposition, or theory is scientific only if it is falsifiable.
- A simple theory applies to more cases than a more complex one, and is thus more easily falsifiable.

What is Simple?

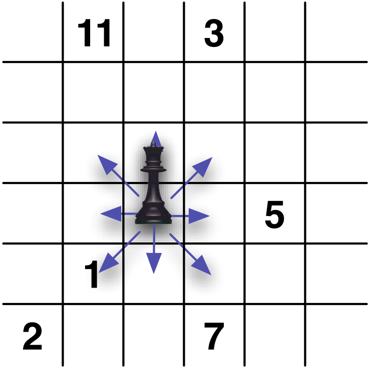
- Occam's razor is not a formal/mathematical objective principle. What is simple for one may be complicated for someone else.
- Idea: Use a computer running programs of some language (say C or Python).
- Encode the data/theory etc. as a computer program.
- The complexity is the length of the shortest program encoding the data/theory.
- "abababababababababab" can be produced by the Python program 'ab' * 10 (with length of 7 bytes); but to encode "aidjendkwasuwemnduen" we need more bytes (as I tried to hit the keyboard [almost] randomly when creating it).

Kolmogorov Complexity

- Use a Turing machine M, and encode data x by the bitstring $< M >$ for the Turing machine itself and the bitstring w which is the program reproducing the data x when run with machine M. $< M >$ cocatenated with w is called a description of the data x.
- The Kolmogorov complexity is the length of the shortest description of x.
- Powerful concept to describe the complexity of data in Algorithmic Information Theory.
- Uncomputable.
- ‘Kolmogorov Complexity’ was invented by Ray Solomonoff (1960) (Matthew effect: ’For to all those who have, more will be given, and they will have an abundance; but from those who have nothing, even what they have will be taken away.’)

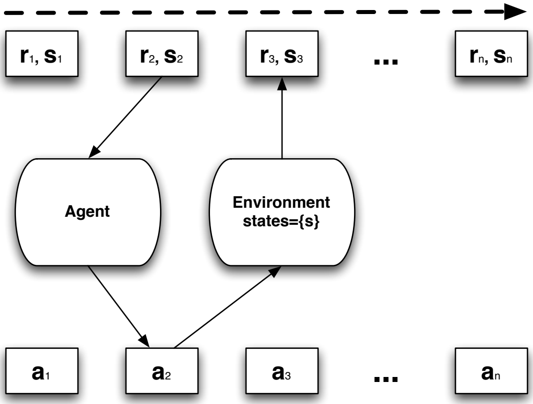
Reinforcement Learning

- Reinforcement Learning is learning from interaction with an environment, from the consequences of action rather than from explicit teaching.
- Agent acts in an environment which is in state s by choosing an action a which leads to a change of state in the environment and a scalar reward r from the environment.



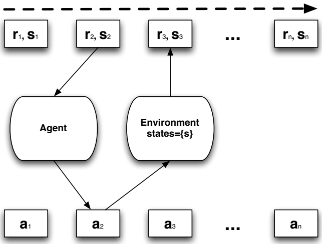
Reinforcement Learning

- Agent acts in an environment which is in state s by choosing an action a which leads to a change of state in the environment and a scalar reward r from the environment.



Reinforcement Learning - Fully observable

- Agent receives some indication of the current state of the environment.
- Chooses action a as output.
- The action changes the state of the environment and the value of this state transition is communicated to the agent via a reward r.
- Goal: Maximise the sum of rewards r.
- How? Learn a policy maximising the sum of rewards.
- Policy : Mapping from states to actions.



Reinforcement Learning

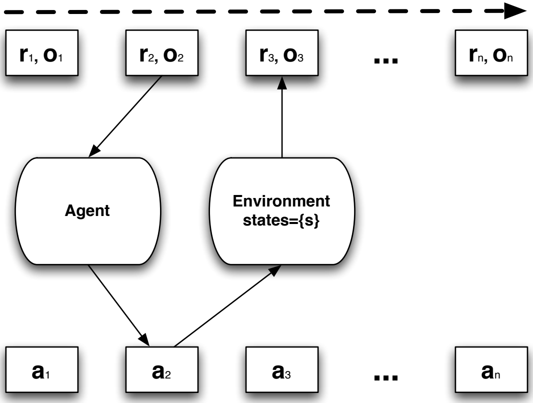
- Generally, the environment is non-deterministic: Taking the same action in the same state on two different occasions may result in different next states and rewards. This results in transition probabilities from one state to another state.
- No supervised output but delayed reward.
- Agent needs to gather information about the states, actions, transitions and rewards.
- Often on-line performance is important (robotics).
- Exploration versus Exploitation
- Temporal Credit Assignment Problem: How do we know whether the action taken now was good? (Wait until the 'end' if an 'end' exists?)
- Game playing
- Multiple agents

Reinforcement Learning - Partially observable

- The agent does not know the state of the environment, but receives observations about it.
- Observations may be noisy.
- Simple strategy: Treat observations as states.
- Leads to State free Stochastic Policies: Mappings from observations to probability distributions over actions.
- POMDP: Partially Observable Markov Decision Process: Use a HMM to learn the hidden states of the environment from the observed data. (Belief states: Probability distributions over the states of the environment.)
- Policy: Now mapping from belief states into actions.

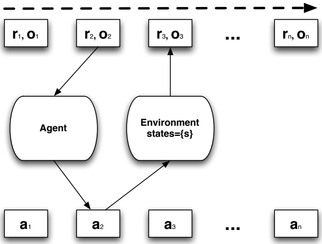
Reinforcement Learning - Partially observable

- The agent does not know the state of the environment, but receives observations about it.



Reinforcement Learning - Partially observable

- A Partially Observable Markov Decision Process is a generalization of a Markov Decision Process.
- A POMDP models an agent decision process in which it is assumed that the system dynamics are determined by an MDP, but the agent cannot directly observe the underlying state.
- Instead, it must infer a distribution over the state based on a model of the world and some local observations.
- Applications include robot navigation problems, machine maintenance, and planning under uncertainty in general.



PageRank

- The problem : How to mechanically rank web pages by importance?
- Page, Lawrence and Brin, Sergey and Motwani, Rajeev and Winograd, Terry *"The PageRank Citation Ranking: Bringing Order to the Web."* Technical Report. Stanford InfoLab, 1999 (All graphics on the following slides taken from this report.)
- Citation from the paper: "To test the utility of PageRank for search, we built a web search engine called Google"
- Size of the Web

year	pages	comment
1998	$150,000,000$	from the paper
2008	$1,000,000,000,000$	one trillion

(2008 according to the offical Google blog `http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html`)

- Not all pages are indexed in search engines. In fact, many are not.

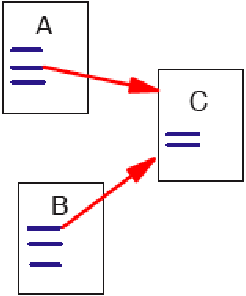
Musing about magnitudes of numbers

- What a difference do 3 digits make?

$$
\begin{aligned}
1,000,000 \text{ seconds} &= 11.57 \text{ days} \\
1,000,000,000 \text{ seconds} &= 31.71 \text{ years} \\
1,000,000,000,000 \text{ seconds} &= 31,709.79 \text{ years}
\end{aligned}
$$

PageRank

- Forward links: A and B are forward linked to C. (out-edges)
- Backlinks : A and B are backlinks of C. (in-edges)
- The World-Wide-Web can be seen as a graph with nodes (pages) and edges (links).
- The graph is not acyclic.
- Note: Can only collect forward links, not backlinks.
- Goal: Calculate the importance of a page from only the link structure. (Don't look at the contents of a page except for the links in it.)



PageRank - Assumptions

1. Many links to a page signal a high importance (rank) for this page.
2. A backlink from an important site counts more than a backlink from a less important site.

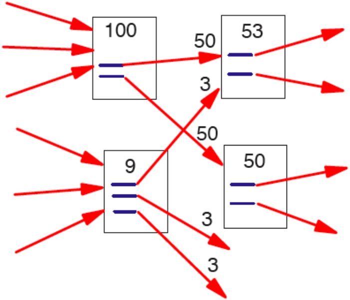
- Combine both of this assumptions: A page has high rank if the sum of the ranks of its backlinks is high.

Simple PageRank - Definition

- Let u be a page. Denote by F_u the set of pages u points to, and by B_u the set of pages which point to u. Denote by $N_u = |F_u|$ the number of out-going links from page u.
- Simple ranking $R(u)$ is then defined as

$$R(u) = c \sum_{v \in B_v} \frac{R(v)}{N_v}$$

where c is a normalisation such that the total rank of all web pages is constant.

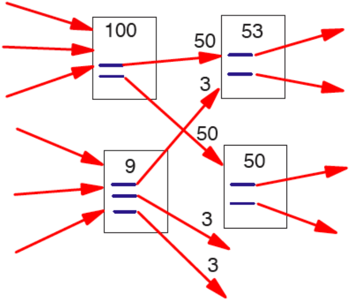


Simple PageRank - Calculating

- Recursive formula

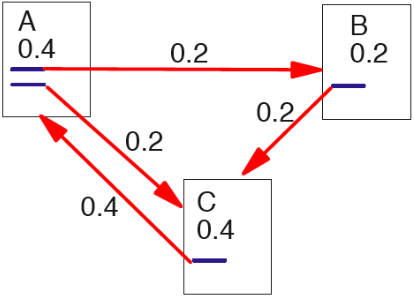
$$R(u) = c \sum_{v \in B_v} \frac{R(v)}{N_v} \qquad \forall\, u$$

- Can be iterated.



Simple PageRank - Converged PageRank

- A simple example after the PageRank calculation converged.



Simple PageRank - Eigenvector Problem

- Create a matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, where N is the number of web pages.

$$A_{u,v} = \begin{cases} 1/N_u & \text{page } u \text{ links to page } v \\ 0 & \text{otherwise.} \end{cases}$$

- Introduce the vector with rankings for all web pages $\mathbf{R} \in \mathbb{R}^N$.
- Then the simple PageRank

$$R(u) = c \sum_{v \in B_v} \frac{R(v)}{N_v}$$

can be written as

$$\mathbf{R} = c\mathbf{A}\mathbf{R}$$

- $\mathbf{R}$ is an eigenvector of $\mathbf{A}$ with eigenvalue $1/c$. (Paper says 'eigenvalue c', but that seems to be wrong.)
- In fact, we are looking for the dominant eigenvector of $\mathbf{A}$.

Simple PageRank - Solved ?

- Nice result: Solve the eigenvector equation

$$\mathbf{R} = c\mathbf{A}\mathbf{R} \dots$$

Simple PageRank - Solved ?

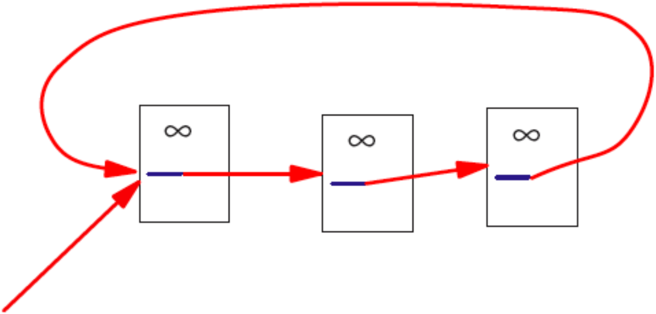
- Nice result: Solve the eigenvector equation

$$\mathbf{R} = c\mathbf{A}\mathbf{R} \ldots$$

- ... for a matrix with $1,000,000,000,000,000,000$ entries.
- Need to iterate.

PageRank - 'Small Problem' with iteration

- Suppose you have two pages, u and v, which link to each other, and have incoming but NO outgoing links.
- During the iterative calculation, these pages will acumulate page rank, but never distribute it. They form a page rank sink.



PageRank - Add a Rank Source

- Introduce a page rank source.
- Let $E(u)$ be some vector over the web pages that corresponds to a source of rank.
- The ranking $R(u)$ is then defined as

$$R'(u) = c \sum_{v \in B_v} \frac{R'(v)}{N_v} + c\, E(u) \qquad \forall\, u$$

  such that c is maximised and for the L_1-norm of $R'(u)$ we have $\|R'(u)\|_1 = 1$.
- In matrix form

$$\mathbf{R}' = c(\mathbf{A} + \mathbf{E}\ \mathbf{1}^T)\mathbf{R}'$$

PageRank - Random Surfer Interpretation

- The process described by the equation for PageRank

$$\mathbf{R}' = c(\mathbf{A} + \mathbf{E}\,\mathbf{1}^T)\mathbf{R}'$$

  can also be interpreted as a random walk on graphs.
- A random surfer clicks randomly on page links.
- But a real surfer will periodically get 'bored' and jump randomly to some other page chosen based on the distribution in E.
- Mostly E will be chosen uniformly. But it can also be used to create 'customised' page ranks.

PageRank - Computation

- 'The computation of PageRank is fairly straightforward if we ignore the issues of scale.' (citation from the paper)
- Let S be almost any vector over web pages (for instance E).

$$
\begin{aligned}
R_0 &\leftarrow S \\
\text{loop}: & \\
R_{i+1} &\leftarrow \mathbf{A} R_i \\
d &\leftarrow \|R_i\|_1 - \|R_{i+1}\|_1 \\
R_{i+1} &\leftarrow R_{i+1} + d\,E \\
\delta &\leftarrow \|R_{i+1} - R_i\|_1 \\
\text{while } & \delta > \epsilon
\end{aligned}
$$

PageRank - Searching

- Given a search item, find all pages which contain the search item.
- Calculate the PageRank for these selected pages.
- Don't need all ranks to provide the user with the first few ranks. But need to find the highest page ranks fast.
- 'Rank merging is known to be a very difficult problem, and we need to spend considerable additional effort before we will be able to do a reasonable evaluation of these types of queries. However, we do believe that using PageRank as a factor in these queries is quite beneficial.' (paper citation)

PageRank - Qestions

- How to deal with the size of the World Wide Web?
- How to update the search engine database?
- How to protect against 'Google bombs'?
- How to protect against manipulation by commercial interests?
- Search engines are an important source of information. Should the exact form of PageRank or any other algorithm added/applied by search engine companies be more transparent to the public?

Envelope Paradox

- You are given two closed envelopes, one of them contains twice the amount of money than the other. You are allowed to open one of them. Then you can choose to keep the money you find, or switch to the other envelope (which could double or half your gain).
- Symmetry: It doesn't matter whether you switch. The expected gain is the same.
- Refutation: With probability $p = \frac{1}{2}$ the other envelope contains double or half the amount. So if you switch, the gain increases by $\frac{1}{2} \times 2 + \frac{1}{2} \times \frac{1}{2} = \frac{5}{4}$.

Envelope Paradox

- You are given two closed envelopes, one of them contains twice the amount of money than the other. You are allowed to open one of them. Then you can choose to keep the money you find, or switch to the other envelope (which could double or half your gain).
- Symmetry: It doesn't matter whether you switch. The expected gain is the same.
- Refutation: With probability $p = \frac{1}{2}$ the other envelope contains double or half the amount. So if you switch, the gain increases by $\frac{1}{2} \times 2 + \frac{1}{2} \times \frac{1}{2} = \frac{5}{4}$.
- Hint: Are the probabilities for the amounts of money equally distributed? In which interval?