Focused Crawler for Topic Specific Portal Construction

Ruey-Lung, Hsiao
25 Oct, 2000

Road Map

Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery (WWW8)
- System Architecture
- Classification
- Distillation
- Evaluation

Using Reinforcement Learning to Spider the Web Efficiently (ICML '98)
- Reinforcement Learning
- Q-learning
- Classification
- Evaluation
Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery

System Architecture

Three major components - Classifier, Distiller, Crawler

Classification (1)
- Bernoulli Document Generation Model

• Generation Model
  • A document \(d\) is generated by first picking a class
  • Each class \(c\) has an associated multi-faced coin
  • Each face represents a term \(t\) and has some success probability \(f(c,t)\), that is the occurrence rate of \(t\) in \(c\).

• Document Generation
  • Terms in \(d\) are generated by flipping the coin a given number of times.

\[
\begin{align*}
n(c, t) &= \left\{ \begin{array}{ll}
\times & \text{if } t \in c[1, L(c)] \\
0 & \text{otherwise}
\end{array} \right. \\
n(c) &= \sum_{t} n(c, t) \\
f(c, t) &= \frac{n(c, t)}{n(c) + L(c)} \\
\text{P}(d|c) &= \left( \frac{n(d,1)}{n(d)} \right) \times f(c, t)^{n(d,1)} \\
\text{P}(d) &= \sum_{c} \text{P}(d|c)
\end{align*}
\]
Classification (2)

- Notation
  - \( C \): concept ontology
  - \( D(c) \): example documents in \( c \)
  - \( C^* \): interested topics
  - \( R_{C}(q) \): relevance measurement given a web page \( q \)

- \( R_{\text{root}}(q) = 1 \) if \( q \) is the root.
- If \( \{C_i\} \) are children of \( C_0 \), \( R_{C_i}(q) = R_{C_0}(q) \)

\[
P(c|d) = P(\text{parent}(c)|d) \cdot P(c|\text{parent}(c))
\]

\[
P(c|d, \text{parent}(c)) = \frac{P(c|\text{parent}(c)) \cdot P(d|c)}{P(d|\text{parent}(c))}
\]

Distillation & Evaluation

- System Goal
  - Find \( V \in D(C^*) \) where \( V \) is reachable from \( D(C^*) \) such that \( V \subseteq R(V)/|V| \) is maximized.
  - Achieve topic distillation mechanism by hub/authority score.

![Graphs showing performance metrics](image)
Reinforcement Learning (1)

• Goal
  - Autonomous agents learn to choose optimal actions to achieve its goal.
  - Learn a control strategy, or policy, for choosing actions.

• Model

  Environment

  Agent

  \[ S_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \ldots \]

  Goal: learn to choose actions that maximize discounted cumulated reward
  \[ r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots \], where \( 0 \leq \gamma < 1 \)

Reinforcement Learning (2)

• Interaction between agent and environment
  - Set \( S \): a distinct states of environment, and set \( A \): a distinct actions that agent can perform
  - Environment responds by a reward function \( r_t = r(s_t, a_t) \)
  - Environment produces the succeeding state \( s_{t+1} = \delta(s_t, a_t) \)

• Markov decision process (MDP)
  - The functions \( r(s_t, a_t) \), \( \delta(s_t, a_t) \) depend only on the current state and action.

• Formulate policy
  - Agent learns \( \pi: S \rightarrow A \), selecting next action \( a_t \) based on state \( s_t \)
  - Policy should lead to maximize cumulative value \( V^\pi(s_t) \).
  \[
  V^\pi(s_t) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots = \sum_{i=0}^{\infty} \gamma^i r_{t+i}
  \]
  \[ \pi^* = \text{argmax} V^\pi(s) \text{ for all } s \]
Q-Learning

- It's difficult to learn $\pi^*$ : \( S \rightarrow A \) directly, because training data does not provide examples of the form \(<s,a>\)
- Agent prefer state \( s_1 \) over \( s_2 \) whenever \( V^*(s_1) > V^*(s_2) \)
- The optimal action in state \( s \) is the action \( a \) that maximizes the sum of the immediate reward \( r(s,a) \) plus the value \( V^* \) of the immediate successor state, discounted by \( \gamma \)
  \[ \pi^* = \text{argmax} \left[ r(s,a) + \gamma V^*(\delta(s,a)) \right] \]
- Corelated measurement \( Q \)
  \[ Q(s,a) = r(s,a) + \gamma V^*(\delta(s,a)) \Rightarrow \pi^* = \text{argmax}_a Q(s,a) \]
- Relation between \( Q \) and \( V^* \)
  \[ V^*(s) = \max_a Q(s,a') \]
- Estimate \( Q \)-value iteratively
  \[ Q'(s,a) \leftarrow r + \gamma \max_a Q'(s,a') \]

Classification & Evaluation

- Mapping Text to \( Q \)-value
  - Given we have calculated \( Q \)-values for hyperlinks in training data
  - Discretize the discounted sum of reward values into bins, place the text in the neighborhood of the hyperlinks into the bin corresponding to their \( Q \)-values
  - Train a naïve Bayes text classifier using those text
  - For each hyperlink, calculate the probabilistic class membership of each bin, the estimated \( Q \)-value of that hyperlink is the weighted average of each bins’ value.

- Evaluation
  - Measurement : \# of hyperlinks followed before 75% target found.
    - Reinforcement Learning : 16% of the hyperlinks
    - Breadth-first : 48% of the hyperlinks