Behavioral Classification on the Click Graph

Martin Szummer and Nick Craswell
Microsoft Research Cambridge
7 J J Thomson Avenue
Cambridge, UK
{szummer, nickcr}@microsoft.com

ABSTRACT

A bipartite query-URL graph, where an edge indicates that a document was clicked for a query, is a useful construct for finding groups of related queries and URLs. Here we use this behavior graph for classification. We choose a click graph sampled from two weeks of image search activity, and the task of "adult" filtering: identifying content in the graph that is inappropriate for minors. We show how to perform classification using random walks on this graph, and two methods for estimating classifier parameters.

Categories and Subject Descriptors

I.2.6 [**Artifical Intelligence**]: Learning

General Terms

Algorithms, Experimentation

Keywords

Click data, classification

1. INTRODUCTION

Behavioral classification involves employing cues from user behavior to classify entities on the web. This is a promising approach for media that is difficult to classify based on content, for example, multimedia such as images, music or mixed web pages, or even plain text when the categories involve high-level understanding (e.g. satire or humor).

Sometimes the class of interest is related to a user activity, so that similar usage corresponds to similar class. We present a method for exploiting user browsing activity to classify web pages. In particular, we take user click behavior, and demonstrate how to employ it to classify pages as adult (inappropriate for minors) or not.

When a user types a query and then clicks a search result, they create a query-URL association. By logging a large number of click events, the search engine can amass a large number of query-URL pairs. These can be viewed as a bipartite graph, where each query is adjacent to one or more URLs and each URL is adjacent to one or more queries.

Our method exploits the structure of this graph. We implicitly look for clusters of nodes (representing queries or URLs) that share clicks to/from each other. We assume that nodes that cluster well are likely to belong to the same underlying category.

The implicit clustering is done via a Markov random walk on the graph. The walk captures the transitivity of class similarity on the graph: if A is co-clicked with B and B is co-clicked with C, then A is also likely to be related to C. Random walks have previously proved to be effective for clustering [3] and semi-supervised learning [2] of data points in metric spaces. They have recently been deployed for ranking results based on click data [1].

This poster shows how to perform classification using the random walk model, and applies it to an "adult" filtering problem, using manually labeled items that were selected because they were difficult borderline cases for a text-based adult classifier. We present evaluation results for the two methods.

WWW 2008, April 21–25, 2008, Beijing, China.
ACM 978-1-60558-085-2/08/04.

2. MODEL

We construct a graph whose nodes $\mathcal{V}$ range over the union of the documents and the queries. The edges $\mathcal{E}$ correspond to user clicks, with weights given by click counts C_{jk}, associating node j to k.

We define transition probabilities $P_{t+1|t}(k \mid j)$ from j to k by normalizing the click counts out of node j, so $P_{t+1|t}(k \mid j) = C_{jk}/\sum_i C_{ji}$, where i ranges over all nodes. The notation $P_{t_2|t_1}(k \mid j)$ will denote the transition probability from node j at step t_1 to node k at time step t_2.

We are now given a set of labeled seed nodes $i \in L$, and wish to classify a given node k. We assume the following model for labeling the node based on the seeds. Interpret the node k as a sample from a random walk that ended at k after t steps. Infer what seed nodes i the walk may have started from; i.e. consider the backward random walk. All starting nodes i have a label parameter $Q(y \mid i)$. Assign a label to the end node k according to the weighted average of its starting node distributions (ranging over all nodes, labeled and unlabeled $L \cup U$).

$$P(y \mid k) = \sum_{i \in L \cup U} Q(y \mid i) P_{0|t}(i \mid k). \quad (1)$$

If all nodes were labeled, the label parameters could be set directly as $Q(y \mid i) = \delta(y, \tilde{y}_i)$. When most nodes have noisy labels or no labels, the label parameters $Q(y \mid i)$ are unknown, and need to be estimated. We shall propose two methods to estimate the parameters.

2.1 Direct Method

In the first method, we simply set $Q(y \mid i) = \delta(y, \tilde{y}_i)$ for

the labeled training nodes $i \in L$ and to 0 otherwise. If one interprets the random walk probability as a distance, this is akin to a nearest neighbor classifier using the random walk similarity measure.

2.2 Average Margin method

The previous method is not robust to noise in the labels of training nodes. Moreover, it may require long random walks at test time, as each test node must be reached from at least one training node to be classified.

Here we formulate a robust method that first estimates parameters for all nodes; the estimation objective maximizes the average classification margin γ_k for labeled training nodes [2], yielding the linear program

$$\max_{\{Q(y|i)\},\{\gamma_k\}} \quad \frac{1}{|L|}\sum\nolimits_{k\in L}\gamma_k \tag{2a}$$

$$\text{s.t.} \quad P(\tilde{y}_k|k) \geq \gamma_k \qquad \forall k \in L \tag{2b}$$

$$0 \leq Q(y \mid i) \leq 1, \tag{2c}$$

$$\sum_y Q(y \mid i) = 1 \qquad \forall i \in L \cup U,\ \forall y. \tag{2d}$$

The parameters $Q(y \mid i)$ are then used as labels in a random walk ending at the node we want to classify. The key result is that the classifier has a closed form solution which involves two rounds of random walk:

$$f(k) = \sum_{i \in L \cup U} \left(\text{sign} \sum_{m \in L} \tilde{y}_m P_{0|t}(i \mid m) \right) P_{0|t}(i \mid k). \tag{3}$$

This procedure can be seen as a first round of estimating labels for all nodes from the few labeled nodes $\tilde{y}_m$, then followed by a second round of assigning smoothed labels to the test nodes. Because of the closed form solution, training is computationally much cheaper than using expectation maximization or traditional large margin criteria.

2.3 Implementation

Since our click datasets are large, we compute the random walks in an efficient way as follows. We represent the transitions as a sparse matrix $\boldsymbol{A}$. For a backward walk, we encode the distribution at step t as a vector $\boldsymbol{q}_j$ with a single unit entry corresponding to the query node j. Then we calculate $P_{0|t}(k \mid j) = [\frac{1}{Z_j}\boldsymbol{A}(\ldots(\boldsymbol{A}(\boldsymbol{A}\boldsymbol{q}_j)))]_k$, in order of the parentheses, and where Z_j normalizes the result to sum to one over k. This is efficient because these matrix operations are sparse.

3. EXPERIMENTAL SETTING

Our experimental setting is image search, using a sample of 2 weeks of image search query-URL click pairs. The URL at each node is actually the web page containing the clicked image. Our labels are whether the page is considered inappropriate for minors due to adult content. The labeled set is not a random sample of the graph. Rather, the labeled set consists of difficult cases where a text-based classifier was uncertain. Our hope is that the behavior-based classifier can resolve some of this gray area.

The graph has 346K queries, 2.5M URLs and 3.2M edges. There are 4700 positive labels and 3000 negative labels. We randomly selected 1000 of the labeled nodes to be training seeds, and 5000 to be testing labels, equally split between the classes.

Table 1: Accuracy for behavioral adult classifiers.

Method	Accuracy
Baseline	73.9%
Random Walk: Direct Method	78.7%
Random Walk: Average Margin	80.2%

We built a baseline classifier that assigned nodes to the closest class, as measured by the number of edges to the nearest labeled node. If two classes were equally close to a node, the tie was broken by choosing the class with the most numerous nodes at that distance. We compared that to random walks with tuning parameters set similarly as in [1], namely backwards random walks with 10 steps, without self-transitions. Table 1 summarizes the results.

Random chance performance is 50% in this experiment. We see that the random walk classifiers significantly outperform the baseline system. The difference between the direct parameter settings and the average margin criterion is fairly small, although still significant.

We noted that the average margin effectively did twice as long a random walk due to its two rounds; therefore we also tried the direct method with twice as many steps (20), which brought its performance up to 79.6%, however, this made it twice as expensive computationally as the average margin method.

4. CONCLUSION

We have described a behavioral classification method for use on the bipartite query-URL click graph. The classifier uses a backwards random walk from each label. Parameters can be set using a direct method or label noise can be reduced using average margin estimation. We compared these to a baseline shortest-path classifier.

Behavioral classification is a general method that should be useful in cases where particular user behaviors correspond to a class of interest. One could also apply these classification algorithms to other Web classification tasks, such as detecting detrimental content (weapons, alcohol, drugs), commercial intent, dominant location of queries, and others.

5. REFERENCES

[1] N. Craswell and M. Szummer. Random walks on the click graph. In *SIGIR Conf. Research and Development in Information Retrieval*, pages 239–246, July 2007.

[2] M. Szummer and T. Jaakkola. Partially labeled classification with Markov random walks. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14, pages 945–952. MIT Press, Jan. 2002.

[3] N. Tishby and N. Slonim. Data clustering by Markovian relaxation and the information bottleneck method. In *Advances in Neural Information Processing Systems (NIPS)*, volume 13, pages 640–646, 2001.