Webspam detection via Semi-Supervised Graph Partitioning

Chris Biemann, Hans Friedrich Witschel
{biem|witschel}@informatik.uni-leipzig.de

Abstract. The aim of our experiments for the WebSpam challenge was twofold: first to explore a mixture of a link graph and a document similarity graph; and second to adapt an efficient graph clustering algorithm to a semi-supervised functionality. The results on the validation sets suggest that page content can be ignored and that the semi-supervised partitioning works very well, especially on the large set.

1 Graph building

In order to build a mixed content-link graph, we first turned the directed link graph into an undirected one. Then, a document similarity graph was constructed in the following way: for each rare term \( t \) that occurred \( n_t < n \) times in the whole collection of web pages, a list of all documents \( d_1, ..., d_{n_t} \) containing \( t \) was constructed. The free parameter \( n \) defines the notion of “rare term” and depends on the size of the collection.

The lists \( d_1, ..., d_{n_t} \) were treated as sentences of natural language and fed to the corpus production engine tinyCC\(^1\) that is usually used for analysis of large text corpora. TinyCC efficiently computes - for all pairs of words that co-occur in sentences - whether the number of joint occurrences deviates significantly from statistical independence.

Applied to the documents of the WebSpam challenge, this produces a list of pairs \((d_i, d_j)\) of documents that co-occur more often than expected in “document sentences” \(d_1, ..., d_{n_t}\), which means that they share many rare terms. For each pair, there is also a significance value \( w_{\text{content}}(d_i, d_j)\), which can be used as an edge weight when interpreting the list of pairs as a document similarity graph.

Mixing of the two graphs was performed by linearly combining edge weights:
\[
w(d_i, d_j) = \alpha w_{\text{link}}(d_i, d_j) + (1 - \alpha) w_{\text{content}}(d_i, d_j).
\]
Since the link graph is originally unweighted, \( w_{\text{link}}(d_i, d_j) \) was set to the average weight of all edges in the content graph for all document pairs \((d_i, d_j)\) in order to make edge weights comparable among the two graphs. The parameter \( \alpha \) was then varied in order to determine how much influence should be given to content and links, respectively.

2 Semi-supervised graph partitioning

Orginally, Chinese Whispers [1] is a parameter-free, randomised graph partitioning algorithm that has linear run-time in the number of edges, allowing the

\(^1\) http://wortschatz.uni-leipzig.de/~cbiemann/software/TinyCC2.html
processing of very large graphs. For the purpose of web spam detection, we employed a yet unpublished semi-supervised version of this algorithm, which is outlined in the following algorithm on graph $G(V,E)$, training $T$.

\begin{verbatim}
for all $v_i \in V$ do
  class($v_i$) = $-1$
end for

for all $v_i \in T$ do
  class($v_i$) = training class
end for

for $\text{it}=1$ to number-of-iterations do
  for all $v \in V \setminus T$, randomised order do
    class($v$) = predominant class in $\text{neigh}(v)$
  end for
end for

return partition $P$ induced by class labels
\end{verbatim}

The algorithm starts by initialising all nodes according to their training classification, all other nodes get label $-1$. Then, for a couple of iterations (we chose 10 in the experiments), all nodes get updated in random order and inherit the predominant class in the neighbourhood. The dominance per class $a$ for node $v$ is computed locally in the neighbourhood $\text{neigh}(v)$ by:

$$\text{dominance} (a, v) = \frac{\sum_{w \in \text{neigh}(v), \text{class}(w) = a} \text{ew}(v, w) \cdot \text{nw}(w)}{\sum_{w \in \text{neigh}(v)} \text{ew}(v, w) \cdot \text{nw}(w)}.$$ 

The initialisation class $-1$ has always dominance 0. Here, $\text{ew}(v, w)$ denotes the edge weight between nodes $v$ and $w$ as given in the graph, $\text{nw}(w)$ is the node weight. In preliminary experiments, we determined $\text{nw}(w) = \frac{1}{\text{degree}(w)}$, i.e. the influence of nodes is weighted down linearly with the number of edges to other nodes. This weighting scheme is motivated by the following: pages that have many outgoing or ingoing links should be less important when propagating classifications w.r.t. spamicity.

3 Results

The results on the two validation sets given for the challenge suggest that:

– Content can be ignored: on the small set results were somewhat inconclusive as to the optimal value of $\alpha$, but $\alpha = 1$ was near-optimal for various values of $\alpha$. On the large set, $\alpha = 1$ was always optimal.

– The semi-supervised graph partitioning seems to work very well, especially on large data sets: the best precision obtained was 88.72% on the small set and 99.58% (184/63814 errors for non-spam, 151/16126 errors for spam) on the large set. Spamicity grading is given by $\text{dominance}(\text{spam}, v)$.

References