TRANSDUCTIVE LINK SPAM DETECTION

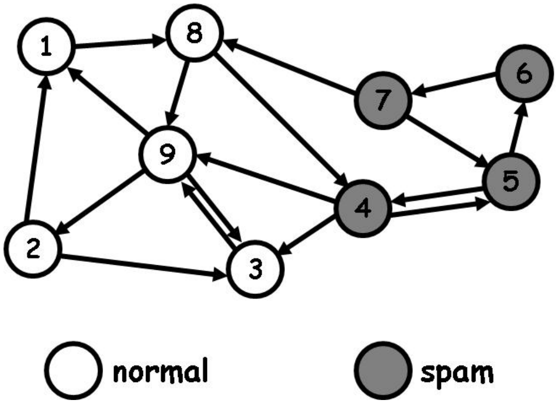
Denny Zhou
Microsoft Research
http://research.microsoft.com/~denzho

Joint work with Chris Burges and Tao Tao

Presenter: Krysta Svore

Link spam detection problem

Classification on a web graph. Those nodes are labeled as two classes, normal and spam web pages.



Basic characters of link spam

- Spam websites generally link to each other, e.g. link farms and link exchange, to boost their link-based rank. In particular, if a website links to spam websites, then it is likely that the website is spam.
- It is unlikely that normal websites link to spam websites. If a website is linked by a normal website, then it is likely that the website is normal.

Spam posts on blogs do not coincide with these two characters.

State-of-the-arts on link spam detection: PageRank based

Restrictions with TrustRank and Anti-Trust or BadRank:

- Do not utilize good and spam examples simultaneously.
- It is unclear what is optimized in those approaches. Consequently, there is no guarantee on their performance.

State-of-the-arts on link spam detection: Supervised Learning

- Machine Learning algorithms with link features
- link features: indegree distribution, outdegree distribution, degree correlation, $\cdots$
- Machine learning algorithms: neural networks, SVMs, Boosting, $\cdots$

Restrictions on Supervised Learning

- Link features for spam are generally extracted from large and popular websites. Thus those websites are not from uniform sampling. The biased sampling leads to potentially high generalization risks.
- We often have very few training examples to utilize because it is costly to label spam by human judgements, while those classical machine learning approaches need a large amount of training examples.
- It might be hard to understand the feature manipulation/combination in the supervised learning process. The understanding is important in spam detection however.

Our methodology: Beyond PageRank and beyond supervised learning

- Cast link spam detection into a semi-supervised learning issue on directed graphs such that we can utilize both labeled and unlabeled examples.
- Develop discrete analogue of classical regularization theory which is widely used in machine learning, e.g. SVMs, and derive our classification algorithm from the discrete regularization.

What is regularization?

- A typical regularization looks like

$$\operatorname*{argmin}_{f \in \mathcal{F}} \left\{ \Omega(f) + C \sum_{i=1}^{m} L(f_i, y_i) \right\}$$

  The first term is for smoothing, and the second term for fitting the given training examples.

- For example, one may define $\Omega(f) = \int \|\nabla f\|^2$, and $L(f_i, y_i) = (f_i - y_i)^2$.

What is regularization? (Cont.)

- Other choices for loss: hinge loss, precision/recall, F_1-score, ROC-Area, $\cdots$
- Other choices for regularizer: kernels, Large margin, spline, hidden layer of neural networks, entropy $\cdots$

Regularization on graphs: let a function change slowly over densely connected subgraphs.

Function spaces on graphs

- Given a directed graph $G = (V, E, w)$, define a random walk on the graph such that it has a unique stationary distribution. Let $p(u, v)$ denotes the transition probability from u to v, and π denotes the stationary distribution.
- Let $c(e) = \pi(e^-)p(e)$. The number $c(e)$ is called the *ergodic flow* on e. It is easy to check that the ergodic flow is a *circulation*, that is,

$$\sum_{\{e|e^-=v\}} c(e) = \sum_{\{e|e^+=v\}} c(e), \ \forall v \in V.$$

Function spaces on graphs (Cont.)

- Let $\mathcal{F}(V)$ denote the set of all real-valued functions on V. A Hilbert space $\mathcal{H}(V)$ over $\mathcal{F}(V)$ can be constructed with the inner product defined by

$$\langle \varphi, \phi \rangle_{\mathcal{H}(V)} = \sum_{v \in V} \varphi(v)\phi(v)\pi(v),$$

where $\varphi, \phi \in \mathcal{F}(V)$.

Function spaces on graphs (Cont.)

- Let $\mathcal{F}(V)$ denote the set of all real-valued functions on V. A Hilbert space $\mathcal{H}(E)$ over $\mathcal{F}(E)$ can be constructed with the inner product defined by

$$\langle \vartheta, \psi \rangle_{\mathcal{H}(E)} = \sum_{e \in E} \vartheta(e)\psi(e)c(e),$$

where $\vartheta, \psi \in \mathcal{F}(E)$.

Discrete operators: gradient

We define the *discrete gradient* $\nabla : \mathcal{H}(V) \mapsto \mathcal{H}(E) \in$ as an operator

$$(\nabla\varphi)(e) := \varphi(e^+) - \varphi(e^-), \forall\varphi \in \mathcal{H}(V).$$

Discrete operators: divergence

As in the continuous case, we define the *discrete divergence* $\operatorname{div} : \mathcal{H}(E) \mapsto \mathcal{H}(V)$ as the dual of $-\nabla$, that is,

$$\langle \nabla\varphi, \psi \rangle_{\mathcal{H}(E)} = \langle \varphi, -\operatorname{div}\psi \rangle_{\mathcal{H}(V)},$$

where $\varphi \in \mathcal{H}(V), \psi \in \mathcal{H}(E)$.

Discrete operators: Laplacian

We define the *discrete Laplacian* $\Delta : \mathcal{H}(V) \mapsto \mathcal{H}(V)$ by

$$\Delta := -\operatorname{div} \circ \nabla.$$

Discrete analogue of regularization

Given a graph $G = (V, E, w)$, the vertices in a subset S have been labeled as `spam` or `normal`. Define a function y with $y(v) = 1$ or -1 if $v \in S$, and 0 if $v \in S^c$. For classifying those unclassified vertices in S^c, we define a discrete regularization

$$\underset{\varphi \in \mathcal{H}(V)}{\operatorname{argmin}} \left\{ \|\nabla \varphi\|^2_{\mathcal{H}(E)} + C\|\varphi - y\|^2_{\mathcal{H}(V)} \right\},$$

where $C > 0$ is the regularization parameter.

Link spam detection algorithm

1. Define a random walk which chooses an inlink uniformly at random to follow. Formally, this random walk has the transition probabilities

$$p(u, v) = \frac{w(v, u)}{d^{-}(u)},$$

for any u, v in V. Let π denote the vector which satisfies

$$\sum_{u \in V} \pi(u) p(u, v) = \pi(v).$$

2. Denote by P the matrix with the elements $p(u, v)$, and Π the diagonal matrix with the diagonal elements $\pi(u)$. Form the matrix

$$L = \Pi - \alpha \frac{\Pi P + P^T \Pi}{2},$$

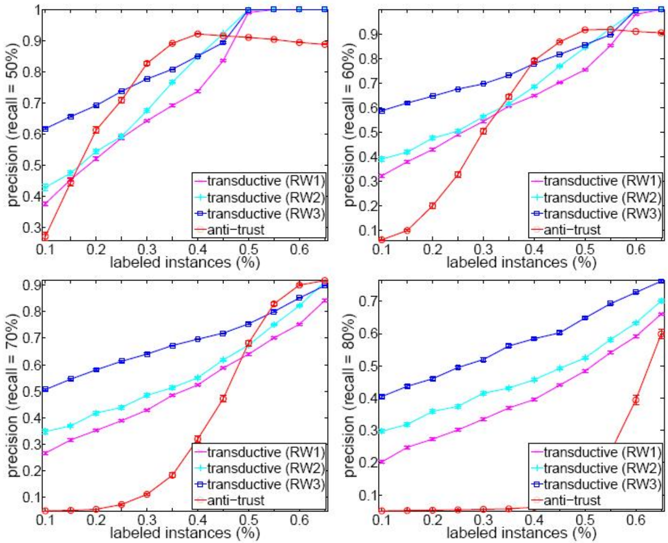
where α is a parameter in $(0, 1)$.

3. Define a function y on V with $y(v) = 1$ or -1 if vertex v is labeled as `normal` or `spam`, and 0 if v is unlabeled. Solve the linear system

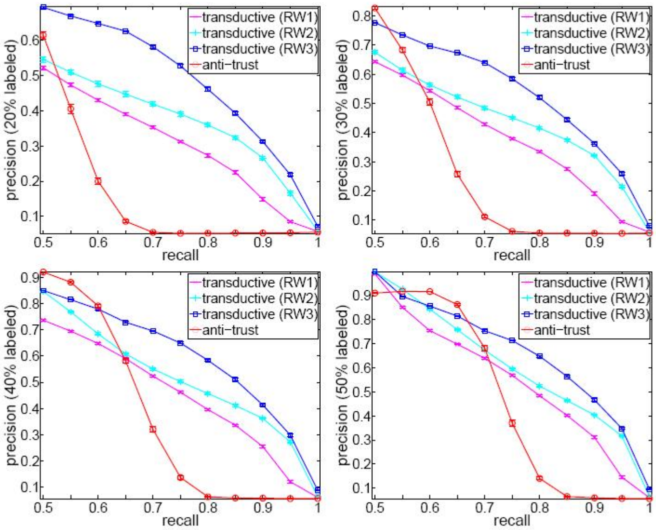
$$L\varphi = \Pi y,$$

and classify each unlabeled vertex v as $\operatorname{sgn} \varphi(v)$.

Experimental results



Experimental results (Cont.)



Conclusion and discussion

- We developed discrete regularization and learning on graphs;
- The basic intuition is to let the classification function change slowly over densely connected subgraphs;
- It is not necessary to extract so-called features from link structures;
- The algorithm be implemented via solving a sparse and symmetric linear system

For combining link and content features in a clean and effective way, please go to our ICML07 paper.

Acknowledgments

We would like to thank Kumar Chellapilla of Microsoft Live Labs for his help on this work.